1991 / AUGUSZTUS

ÁRA: 196 FT

ALAPLAP

MIKROSZÁMÍTÓGÉP MAGAZIN MÁGNESLEMEZ MELLÉKLETTEL



„Világraszóló" ASCII-utód

PC-planetárium

Tiszta Fred, a kapitány

Minél rosszabb — annál jobb

A HÓNAP TÉMÁJA:

VÍRUSVADÁSZAT

Magyar shareware-katalógus

Kártyaparti melankóliában szenvedőknek

Hannibál azelőtt portás volt!

„A vírusok nem kártyakompatibilisak"

A MÁGNESLEMEZEN:

- Hogy a vírus ne visítson!
- A SoundBlaster programozása
- Problémától a programig
- Struktúra-primitívek
- TEGL-demó

A minőség megéri az árát!

Amikor elkezdtük, nemcsak mi, hanem a számítógép-vásárlók többsége is kezdő volt. Valamennyien elkövettük azt a hibát, hogy elhittük: érdemes a legolcsóbb, még éppen működő gépekkel foglalkozni. Az évek során azonban rengeteg tapasztalatot szereztünk, és ma már tudjuk: a számítástechnikában sincsenek csodák. A minőségnek és a megbízhatóságnak ára van, s ha ezen spórolunk, az mindkettő rovására megy.



A fejlett piacgazdaságú Nyugaton azt tartják: a nagyon olcsó, bóvli termék a szegények adója. Aki mégis megveszi, rövidesen tapasztalja: rossz üzletet kötött.

A mindenáron való árcsökkentés ugyanis odavezet, hogy egyre gyengébb lesz a minőség, míg végül elérkezünk arra a pontra, amikor ez már nem éri meg a még oly olcsó árat sem. Mi nem akarjuk, hogy ügyfeleink később azt mondják: átvertük őket, egy vackot sóztunk rájuk, ezért inkább nem is forgalmazunk a legalsó árkategóriába tartozó számítógépeket és nyomtatókat. Úgy gondoljuk: egy átlagosan jó minőségre minden vevőnek joga van. Mi ezért nem tekintjük jelszónak, hogy nálunk **a minőség mindig megéri az árát.** Ez több annál, **ez a mi üzleti filozófiánk.**

De ez az ár nem feltétlenül magas — sőt! Először is az **IR** tudja, hogy sokféle igény, feladat és természetesen sokféle pénztárca létezik. Ezért többféle áron, többféle minőséget kínál, az átlagos felhasználónak megfelelő, olcsó „standard" típustól a profiknak szánt, előtesztelt alkatrészekből szerelt, 72 órán át járatott „extra" kategóriáig, amelyhez kétéves garanciát adunk.

De nemcsak ezért lehetséges, hogy az **IR Szervizben** mindenki megtalálja az igényeinek és anyagi lehetőségeinek megfelelő számítógépet. Segíti ezt az is, hogy az **IR** ma már akkora forgalmat bonyolít le, hogy szállítói megadják neki a legjobb vevőnek járó árkedvezményeket. Az **IR** pedig ezeket **a kedvezményeket maradéktalanul átadja** a hazai számítógép-felhasználóknak.

Így lehetséges, hogy egy kiváló minőségű 286-os AT, 40 Mbájtos merevlemezzel, 1 Mbájt RAM-mal, floppyval és nagyfelbontású, színes VGA monitorral **10%-kal olcsóbb az átlag hazai árnál.** Hasonló a helyzet a hordozható számítógépeknél is. De ugyanilyen jók az árai az egyedi igények alapján összeállított konfigurációknak és hálózatoknak is.

És ez még nem minden. **A nyári hónapokra időszakos számítógépvásárt hirdet az IR Szerviz.** Ennek során — konfigurációtól függően — akár **50 000 forintos árkedvezményt** is kapnak a vásárlók. Az oktatási intézmények pedig további nagyon jelentős, **25%-os kedvezményt** élveznek.

Természetesen a választás joga az Öné.
Keressen fel bennünket Budapesten,

IR

IR Szerviz
VII. kerület, Kis Diófa utca 6.
Tel.: 121-3230, 141-0880



írjon levelet, vagy egyszerűen hívjon telefonon, a 121-3230 vagy 141-0880-as számokon, és mi segítünk, hogy kiválassza az Önnek megfelelő számítógépet. **Bármit választ, nyugodt lehet: olyan minőséget kap, amely biztosan megéri az árát!**

ALAPLAP

Mikroszámítógép magazin mágneslemez melléklettel

Megjelenik havonta

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

Szerkesztőségi titkár:
Sziebig Andrea

A mágneslemez melléklet és a Közkincs szerkesztője:
Verebély Pálné

A Lemezkalauz szerkesztője:
Vékony Tamás

Szerkesztőség, kiadó és hirdetésszervezés:
XI., Karolina út 17.
Budapest 1251
Telefon: 185-1584
Fax: 185-2221

Felelős kiadó:
Sebestyén Ilona igazgató
Cédrus Informatikai Rt.

Nyomdai előkészítés:
Tipoprint Kft., Budapest
Nyomtatás:
Zalai Nyomda, Zalaegerszeg
Felelős vezető: Galla József

Terjeszti a Magyar Posta. Előfizethető a hírlapkézbesítő postahivataloknál és a Posta Hírlapelőfizetési és Lapellátási Irodájánál (XIII., Lehel u. 10/a, Budapest 1900), vagy átutalással a 215-96162 pénzforgalmi számra.
Példányonkénti ár: 196 Ft
Évi előfizetési díj: 2 352 Ft

Külföldre terjeszti a Kultúra, Pf. 149, Budapest 1389

HU ISSN 0865-9788

A HÓNAP TÉMÁJA: VÍRUSVADÁSZAT

(Írta és összeállította: Kis János)

TÉMABŐVÍTŐ

NYÚZÓPRÓBA

KÖZKINCS



SOLARSOFT LEMEZKALAUZ

SZÖVEGELŐ

ABAKUSZ

KILÁTÓ

GÉPRAJZ

ALAPJÁRAT

FOGÓDZÓ

PROGRAMOZÁSTECHNIKA

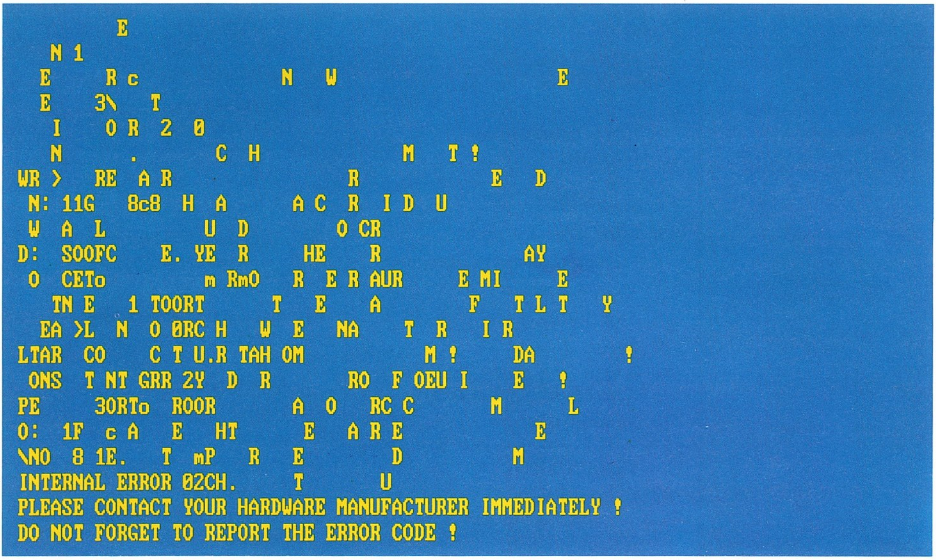
VISSZACSATOLÁS

PALETTA

MÁGNESLEMEZ MELLÉKLET ❑

Hogy a vírus ne visítson!
A SoundBlaster programozása
Problémától a programig
Struktúra-primitívek
TEGL-demó

Címlapképünk a Mathematica, Inc. reklámjából



Mit tehetünk?

A kérdést sokan felteszik, miután elolvassák valamelyik szaklapban az aktuális vírusveszélyre felhívó cikket. Az ilyenkor adott tanács, mely szerint csakis hivatalosan megvásárolt programot használjunk, még a nálunk jóval fejlettebb informatikai kultúrával rendelkező országokban sem követhető. Akkor mit tehetünk programjaink, adataink épségének megóvásáért? Ehhez szeretnénk némi segítséget adni.

Akinek még nem lenne meg, először is be kell szereznie a McAfee-féle SCAN/CLEAN/VSHIELD programrendszert. A jelenleg forgalomban lévő szabadszoftverek közül ez a legkorrektebb. Vírusjelzéskor az /M, azaz memóriavizsgáló opciót használva sok vakriasztás fordul elő, így azt akkor kell komolyan venni, ha a program egy állományban ugyanezt a vírust jelzi.

Alkalmazhatjuk a a TBSCAN programot, illetve annak rezidens változatát. Ennek a programnak a kibocsátója — legalábbis a TBSCAN.DAT állomány aktualizálásakor — sajnos tudatos piaci manipulációt is végez, mert kihagyja azokat a vírusokat, amelyek átmennek az általa forgalmazott Thunderbyte kártyán. (McAfee-nek ez nem áll érdekében.) Ennek ellenére a rezidens kereső és a TBSCAN.DAT állomány megfelelő kiegészítések után viszonylag jó biztonságot nyújt.

A Norton Antivirus programcsomag az eddigi három felfrissítéssel együtt is gyenge, detektálásra csak alkalomszerűen és másodlagosan használható. Hasonló a helyzet a Carmel Software TNT antivírus programjával. A nálunk elterjedt angol nyelvű variáns rosszul irtja a hazai Yankee Doodle-változatokat, míg az újabb, német nyelvű verzióban ezt már kijavították.

Néhány hasznos tanács a vírusveszély csökkentésére:

1. Még a floppyn ellenőrizzünk minden új programot a McAfee-féle SCAN /A /X opciójával, majd utána a TBSCAN programmal.

2. A gépben legyen rezidens vírusvédő program, például a TBSCAN megfelelő géptípushoz írt verziója, hogy az önkicsomagoló installálásoknál ellenőrizhesse, nem fertőzöttek-e a kipakolt állományok.

3. Mindig legyen kéznél a szükséges meghajtókat, az AUTOEXEC.BAT-ot, a víruskereső programokat tartalmazó, írásvédett (leragasztott) boot-lemezünk.

4. A másolásvédett programok használatát a lehetséges maximális mértékben kerüljük el. Különösen ne használjunk másolásvédett vagy korlátozott számú installációra jogosító operációs rendszert vagy hálózati szoftvert.

5. Vírusfertőzéskor lehetőleg a fertőzetlen formában floppyn meglévő szoftvereket installáljuk újra a gépre. Ha nincsenek, csak akkor kíséreljük meg a fertőzés „levakarását".

6. Hardveres vírusvédelemhez várjuk meg a valódi biztonságot nyújtó, kipróbált vírusvédő kártyák kifejlesztését.

7. Figyeljünk a szakirodalomra, de vegyük komolyan a szakma „nem hivatalos" információs csatornáin keresztül érkező figyelmeztetéseket is.

8. Ne takarékoskodjunk a floppyval. Minden fontos programunkról és adatállományunkról legyen elmentett példányunk, de erre ne a DOS backup programját használjuk, hogy ne legyen DOS-verziótól függő a mentés.

9. Ne játsszunk a vírusokkal, mert abból kellő hozzáértés nélkül kiszámíthatatlan károk keletkezhetnek.

10. Ha bekövetkezik a fertőzés, és a rendelkezésünkre álló programok nem segítenek, feltétlenül hívjunk szakembert, aki a vírustalanításban gyakorlattal rendelkezik.

Intelligencia erkölcs nélkül

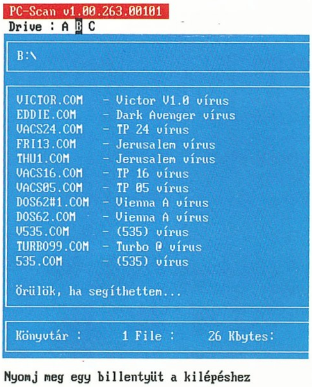
Merre tartanak a vírusok?

Vajon meddig terjedhet a vírus intelligenciája és erkölcstelensége — tehetjük fel ismét a kérdést, miután megjelentek a második és harmadik generációs, eredetileg hadviselési céllal kifejlesztett vírusok.

Egyesek szerint a jövő a mutáló vírusoké. Ha ugyanis megfelelően megírják a víruskódot, akkor elérhető, hogy a vírus folyamatosan változtassa bináris kódját, miközben funkciói ugyanazok maradnak. Ugyancsak „ígéretesek" a bináris harci gázok mintájára létrehozott többkomponensű vírusok. Ezek külön-külön csak ártalmatlan, öncélú játéknak tűnő kódrészletek, és akkor válnak aktívvá és rombolóvá, amikor egymással találkoznak. Minden mutáló vírus „őstípusa", a Whale (magyarul Bálna) több vírussal is képes ilyen kapcsolatra lépni. Ha találkozik a Fish-6 vírussal, akkor azzal átmenetileg összeolvad, és a konglomerátum halakról elnevezett vírusok tömegét szüli meg, majd ismét szétválik.

Az új vírustípusok képesek visszafejteni (debugolni) a megtámadott rendszert, és beülni annak a tetejére, a rendszerállományok elé vagy más rejtett helyre. De hasonló módon megtehetik, hogy másik vírussal találkozva azt is visszafejtsék, és bizonyos rutinjait beépítsék önmagukba. Ilyen rendszerek már ma is írhatók, de a kód mérete még túl nagy.

A vírusterjesztés egyik újdonsága, hogy a közkedvelt programokba linkelnek tömörített vírusokat, s azok adott

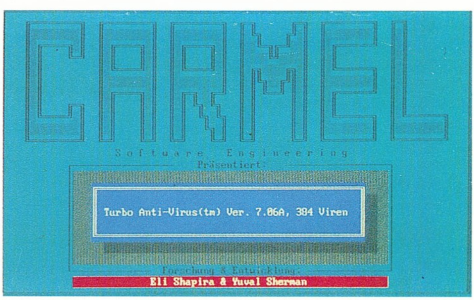


feltételre bontják ki magukat. Nagy a kísértés ennek a technikának a másolásvédelemben való felhasználására is. Németországban jelent meg például a „Raubkopieren" programvírus, amely valószínűleg egy játék másolásvédelméből indult útjára.

A hazai eredetű Töltögető is felvillantott egy új módszert a második generációs vírusok írásában. Ennek lényege, hogy elindítanak egy önmagában értelmetlen vírust, amely felteszi a megfelelő rutinokat a merevlemez rejtett helyére, vagy floppynál az utolsó, szabványosan nem használt sávra. Utána egy másik, önmagában szintén értelmetlen vírus hozzáadja ehhez a kódhoz a betöltő részt. Ha pedig az utolsó pályát nem szabványosan formázták — ahogy azt a másolásvédelmek is szokták —, akkor rendkívül bonyolult és jól eldugott rendszer helyezhető el ott.

A vírusírás és a másolásvédelem egyébként is kéz a kézben jár egymással. Nyugat-Európában mind gyakrabban hoznak forgalomba olyan nagy értékű programrendszereket, amelyek csak megadott futásszámig üzemképesek, utána „problémák lépnek fel". Ez gyakorlatilag az IBM-nek korábban a nagygépeken alkalmazott taktikáját ülteti át a PC világába. (Akkoriban a gép operációs rendszerét kellett a felhasználónak a zavartalan használat érdekében újra meg újra megvásárolnia.) Ezen irányzat európai terjedését erősítik azok a hardverkulcs-forgalmazó cégek is, amelyek kiszorulnak az amerikai piacról. S miközben a vírusírás és a programfeltörés „iparággá" válik, a tisztességes szoftverkereskedelmet sem az eladók, sem a vevők oldaláról nem könnyű ösztönözni. Ennek az alagútnak egyelőre nem látszik a vége.





Pokoli panoptikum

A vírusírók egyre inkább gondolnak a vírusellenes szoftverekre: vagy azok ellen írnak vírusokat, vagy vírusaik elmentés nélkül felülírják az állomány egy részét. A terjedési és pusztítási szakasz között csökken az időintervallum. Ugyanakkor új programozási technikák jelennek meg, s közforgalomba került az első vírusgeneráló programcsomag is. A vírusok között pedig egyre több a trójai program, bár azok szerzői nem mindig viszik el szárazon, mert büntetőjogilag könnyebben felelősségre lehet vonni őket.

A bélyeggyűjtés egyik fő módszere a csere. A vírusgyűjtők most kezdik ugyanezt alkalmazni. De a bélyeget nem szokták „átírni", mielőtt továbbadják, a vírusokat viszont legalább egy-két bájt erejéig igen, mert az növeli „csereértéküket". Így azután a vírusmutációk száma dinamikusan nő.

Az alábbiakban a vírusok eme pokoli panoptikumának néhány újabb tagját mutatjuk be. Többségük a környező országokból származik, így félő, hogy rövidesen nálunk is elterjednek.

Új generáció

Mini-45

Az elektronikában a miniatürizálás a fő irányzat. A vírusírók is versenyeznek, hogy ki tudja megírni a legrövidebb működő vírust. Az Alaplap egyik régebbi számában már írtunk egy 144 bájt hosszú vírusról, amely annak idején csúcs volt. Kiderült, hogy a vírus orosz eredetű, s idén nyár elején ugyanott megjelent a még rövidebb Mini-45 vírus, ami nevének megfelelően már csak 45 bájt hosszú. A vírus nem rezidens, a COM állományokat fertőzi meg, és a Scan 77 nem ismeri fel.

Tequila

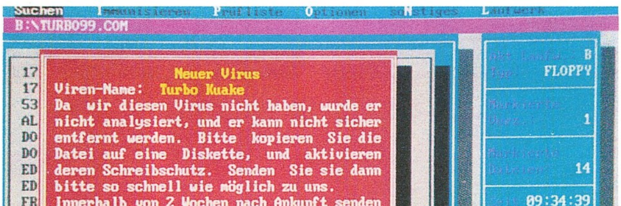
Nagy vihart kavart a Tequila vírus megjelenése és terjedése. Ez a fájlvírus a merevlemez nullás pályájára installálja magát a fájlból. A vírus működésének pontos leírását később, a visszafejtés után közzétesszük, de érdekes a vírus íróival kapcsolatos hír is.

1991. május 20-án a svájci rendőrség letartóztatott egy 18 és egy 21 éves fiatalembert a Tequila vírus írásának és terjesztésének alapos gyanújával. Az esemény a svájci St. Hausen városban történt. A fiúk nemcsak megírták a vírust, hanem terjesztői is voltak. Az egyik fiú édesapja ugyanis szabadszoftverek, játékprogramok disztribútora volt Svájcban. A Tequila vírust így sikerült a fiúknak a shareware programok közé becsempészni és Európában elterjeszteni. A rossz vicc következményeként az apa elvesztette vállalkozását, és a kártérítések miatt anyagilag tönkrement.

A vírusban lévő szöveg:
Welcome to T.TEQUILA's latest production.
Contact T.TEQUILA/P.o.Box
543/6312 St'hausen/Switzerland.
Loving thoughts to L.I.N.D.A
BEER and TEQUILA forever !

Patricia

A vírust — neve alapján — Patricia Hoffmannak szánták „ajándékul" az olasz vírusírók. Patricia Hoffman készíti ugyanis McAfee cégénél a vírusok leírását tartalmazó adatbázist. A vírusban lévő szöveg alapján a vírus a memória farokpointerének módosításával a memória tetejére beépülő destruktív rezidens vírus. Üzenetei:
Is today Friday? (Y/N)
Sorry but on Friday I wish not to work!!
You are untruthful!!
For punishment I'll format your HD Fat!!
This virus was written in Italy by Cracker Jack 1991 IVRL
All rights reserved, please don't crack this virus!!



Special message to Patricia Hoffman:
I love you!!!!!!!! SmackSmack!!
Can you give me your telephone number???
Ciao bellissima!

A vírus aktivizálódása után tönkreteszi a lemez FAT tábláját. Ezek után más már nem segít, csak a DOS formázó parancsa.

Az utóbbi hónapok tapasztalatai alapján a Patricia jellegű destruktív vírusok és a trójai programok készítése és terjesztése erősödik. A vírusírók arra törekszenek, hogy munkájuknak alapos nyoma maradjon. Az ilyen vírusok ellen az időszakos vírusellenőrző megoldások nem védenek.

Kártyajáték

A szoftveres vírusvédelemnél talán hatékonyabb lesz majd a vírusvédelmi kártya. TG-kártyánk tesztelése során meggyőződhettünk arról, hogy a jelenleg mintegy 400 víruscsaládot felölelő tesztállományból egyetlen vírusnak sem sikerült a kártyát megkerülnie, szabadon elindíthattuk a vírusokat, és azok nem fejtették ki romboló hatásukat, viszont tanulmányozhattuk működési mechanizmusukat és aktivizálódási képernyőjelenségeiket. Ezek közül íme néhány:

1381

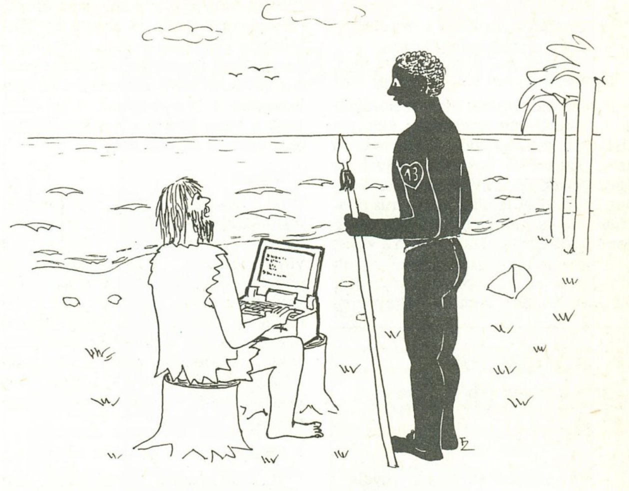
Elindítása után a vírus a képernyőn lévő szöveget szétrobbantja. A kép úgy néz ki, mintha a monitor alsó sorában egy bomba robbant volna. A jelenség után a vírus kéri a felhasználót, hogy okvetlenül vegye fel a kapcsolatot a hardver szállítójával. (Lásd 2. old.)

Önállósult vírusszolgálat

Az Ázsió-Microtrade (Ázsió-Viki) Kft.-ből kivált szakemberek a továbbiakban Safe Kft. névvel önálló vállalkozást indítottak, és abban végzik tovább vírusvédelmi, adat-helyreállítási tevékenységüket, kibővítve azt a hozzá kapcsolódó adatbiztonsági feladatokkal. Végleges telephely és telefonvonal hiányában még nem tudták újraindítani a Virnet Hungary BBS-t. Ideiglenes elérési helyük:

Safe Kft.

Szegedi Imre
XIII., Gidófalvy u. 31.
1134 Budapest
Telefon: 140-7681



Robinson: Menj arrébb, nehogy megint megfertőzd a számítógépet!

Virus 101

A vírus első elindítása után egy nagy piros ablakban megjeleníti az alábbi copyright szöveget, majd folyamatosan elkezdi pörgetni a floppymeghajtó motorját. Ha a floppymeghajtóba nem teszünk lemezt, akkor a rendszer újraindításáig folyamatosan így fog működni. Amint floppylemezt tettem a meghajtóba, a vírus azonnal meg akarta fertőzni a COMMAND.COM programot.

VIRUS 101
Copyright 1990
Patrik A. Toulme
Suite #104
2007 N. 15th Street
Arlington, VA 22201
(703) 836-9340 Ext. 225

This file has been safely infected by VIRUS101 — the safe, education virus utility.
This copy furnished to XY.....

Ha a vírus valamilyen állományt megfertőz, akkor azt követően egy összeomló képernyővel újraindítja a rendszert. A vírus betöltése hangeffektusokkal is jár. A vírus bonyolult kódolási formát tartalmaz, és ellenőrzötten terjesztették, az egyes elengedési ponton saját nevüket (XY ...) kódolva be, hogy ezzel is ellenőrizzék a vírusok áramlását.

Iraqui Warrior

Az iraki háború idején és utána sorra jelentek meg a háborúval kapcsolatos számítógépvírusok. Ilyen például az Iraqui Warrior vagy a Saddam vírus több változatban is. Az Iraqui Warrior elindítása után a következő szöveg jelenik meg:

I come to You from The Ayatollah!
(c) 1990, VirusMaters
An Iraqui Warrior is in your computer...

Finger

A vírus egyik célpontja a ZIP állomány, amelynek részeit felülírja. A vírus elindítása után a következő üzenetet írja ki:

Cannot remember what I was doing!!
Insert fingers is ears and reboot please

ARF

Az egyszerű szoftveres védelem ebben az esetben nem sokat ér, mert a vírus BIOS szintű közvetlen sávformázást végez, s már csak konstatálni tudjuk a kárt. Tehát a vírus bejutását kell megakadályozni, nem pedig az utólagos gyógykezelésben reménykedni. A TG vírusvédelmi kártya mellett nyugodtan lehetett futtatni a vírust, tiszta szoftveres megoldással ez nem ment volna. Amikor a vírus a merevlemez egyik sávját formázta, akkor a következő üzenetet jeleníti meg:

Arf Arf Got you!
-- RABID '90

Ezt követően már késő a víruskereső és vírusölő programokat futtatni.

Hymn

Van belőle jócskán Magyarországon is. Legalább három mutánsa van. Az első verziója a legrövidebb, a következő kettő valamivel hosszabb, mivel az egyik önkódoló, a másik pedig anti-debug funkciókat tartalmaz. A vírus elindítása után semmilyen feltűnő jelenség nem tapasztalható. Ha azonban a vírust a Magnitogorsk 2048 vírussal együtt indítottam el, az operációs rendszer lefagyott. A reset gombot megnyomva a számítógép belső modemének hangszórója elkezdett zenélni, és az operációs rendszer bejött. A zene és a DOS betöltése független volt, a számítógép hangszórója a rendszerindításnál sípolt. Ezt követően zenei aláfestéssel tudtam használni a számítógépet. Úgy tűnt, hogy a vírus kibírta a rendszerindítást és a modemben bújt meg.

Mirror

A vírus nem hosszú, de gyilkos. Aktivizálódása során a képernyő betűit tükrözi. (Az Ifabón és a BNV-n ez a vírus „tesztelte" a TG vírusvédelmi rendszert.) A Norton Commander ablakait látványosan cserélgeti. Több felhasználó panaszkodott, hogy valami nagyon gyorsan legyalulta merevlemezüket, 5 másodperc alatt nullázta a 80 Mbájtos harddiszket — a rajta lévő vírussal együtt. Milyen vírusölő program segít ilyenkor?

Shake

A vírus semmi különlegeset nem tartalmaz. Aktivizálódása után a következő üzenetet írja ki:

Shake well before use!

Sorry

A vírus COM állományokat fertőz. Az elnézést kérő német nyelvű üzenet tartalma miatt lett a neve Sorry. Aktivizálódása után kiírja:

Bitte gebe den G-Virus Code ein:

Egy szám megadása után (valószínű a rossz válasz) pedig jön a sajnálkozás:

Tut Mir Leid !

Brain Slayder

A vírus nemcsak COM, EXE és overlay programokat támad meg, hanem szétveri az adatállományokat, és a FAT fájlbejegyzésben láncolási hibát csinál. Az ilyen típusú vírusok pusztítása után egyszerűen nincs mit helyreállítani. (A FAT láncolási hibákra esetleg lehet orvosság.)

V516

A V516 vírus COM programokat megfertőző szokásos rezidens vírus. Érdekessége, hogy a programok által lekérdezett DOS-verziószámot állítgatja. Szélsőséges esetben 0-ás DOS-verziót ad vissza, és a DOS programok sem futnak a hibás verziószám miatt. A Sysdoki 4.xx vírusölő program sem tudja kiirtani, mivel floppylemezről csak keres, s a memóriából nem öli a vírust. A merevlemezen rossz DOS-verziót kap vissza.

Leprosy

A Leprosy vírus első példányát 1991 májusában fedezték fel. Azóta több mint 4 mutáns forog közkézen. A vírus hossza 321 és 370 bájt között van. Fertőzése során egyszerűen felülírja a fertőzött programot. A McAfee-féle dokumentáció is azt javasolja, hogy töröl-

Különleges DOS-verziókhoz: F-Prot

Fridrik Skulason az izlandi egyetemen, Reykjavíkban egy egyszemélyes „szoftverház" tulajdonosa. Mellékesen az angol rendőrségnek segít a vírusírók lefülelésében, s ugyanakkor szerkeszti a Virus Bulletin című, igen zárt körben terjesztett folyóiratot. F-Prot 1.14 szoftverének kezelhetősége meg sem közelíti a McAfee-féle programokét. Ami miatt mégis figyelmet érdemel, az egyetlen tulajdonsága, hogy a McAfee-féle programokkal ellentétben a partíciótáblás és bootvírusok eltávolítására abban az esetben is kiválóan alkalmazható, ha nem a szabványos PC-DOS vagy MS-DOS operációs rendszert használjuk. Vírusok ellen kezeli a Compaq 3.31 és a Tandon 3.30 DOS-verziók extra méretű partíciós tábláját és boot-rekordját, de akkor sem jön zavarba, ha DR DOS 5.0-val formáztuk meg lemezünket.

Az F-Prot csomagból az F-FCHK.EXE program a memóriát nézi végig, míg az F-SYSCHK.EXE az egyes állományokat. Az /AUTO opció esetén a helyreállítást is elvégzi. Az F-INOC.EXE program a bootvírusok ellen alkalmazható.

A szoftverhez mellékelt dokumentációs állományok jól eligazítják a felhasználót. Újabb verziói viszonylag ritkán jutnak el Magyarországra. Kissé ugyan bonyolultan kezelhető, de a magyar viszonyok között is jól alkalmazható. Felismeri például a magyar Töltögető és Turbo Kukac vírusokat is.

Murphy és a vírusok

Vírusok alapvető ismérve
Vírusok elvben csak „garantáltan fertőzésmentes" program- és rendszerlemezeken terjednek.

Vírusok időzítési ismérve
Akkor kapsz vírust, amikor a legkevésbé van rá szükséged.

Vírusok társasági ismérve
Veszélytelen és könnyen eltávolítható vírusokat mindig csak mások szereznek.

Vírusok földrajzi ismérve
Mindig olyan vírusfajta telepszik a gépedbe,
— amelyről a szakértők azt állítják, hogy hazánkban még nem lépett fel,
— amelyhez nincs víruskereső program,
— és amelyik rombolásban és gonoszságban minőségileg újat mutat fel.

A vírusok mennyiségi ismérve
Mindig eggyel több vírusod van, mint éppen gondolod.

A vírusok minőségi ismérve
A számítógépedet megtámadó vírus mindig azokat az állományokat fertőzi meg, amelyekről nem készítettél backupot.

A teljes rekurzív vírus-ismérv
A víruskereső program számítógépeden valamennyi szöveg- és programállományodat helyreállíthatatlanul tönkreteszi, csupán a vírust kíméli meg.

(Joachim Graf: Murphys Computergesetze. Markt & Technik Verlag, 1990)

jük a fertőzött programot. (Nem segíthet semmilyen szoftveres vírusölő program, hiszen az állomány elejét a vírus felülírta.)

Leprosy C

A Leprosy C változat EXE programokat megtámadó és azokat felülíró vírus. Az aktuális könyvtárban lévő első három nem fertőzött állomány első 321 bájtját felülírja. A tönkretett programot elindítva a vírus által felülírt rész a szerzőt ajánlja. A vírus fertőzése során helyreállíthatatlanul roncsol.

Leprosy D

A Leprosy D a COM, EXE programokat felülíró vírus, beleértve a COMMAND.COM-ot is. A megfertőzött gépet újraindítva a „Bad or Missing Command Interpreter" üzenetet kapjuk vissza, ami a COMMAND.COM tartalmának felülírása miatt történik. Ez a vírus is a fertőzött állomány első 321 bájtját írja felül. Többi reakciója megegyezik a Leprosy C vírusével.

A Leprosy vírusok fertőzése során a fájlok mérete nem változik meg. Ha a fertőzendő fájl mérete esetleg kisebb, mint a vírus hossza, akkor a fertőzés után az állomány felveszi a vírus hosszát. A vírus a fertőzés során a fájl dátumát/idejét megjelöli, hogy erről ismerje fel a többszörös fertőzés elkerülése végett. A vírus azonosító bájtjai:

„56 33 F6 E8 51 00 0B C0 74 0A E8 18 00 46 FE 06" Leprosy C

„56 33 F6 E8 4D" Leprosy header

Exterminator

Az Amiga-, Atari- és az Apple-tulajdonosok már ismerik a Lamer Exterminator nevű vírust, amelynek PC-verziója 1991 májusában jelent meg. Ez új generációs, felülíró típusú. Nem rezidens. Fertőzése során az aktuális könyvtárban lévő COM programok első 451 bájtját írja felül, beleértve a COMMAND.COM programot is. A vírus a nevét a vírusban lévő szöveg alapján kapta.

Exterminator 1.0 — (c) by Cracker Jack 1991 (IVRL)

Italian Virus Research Laboratory (C) 1990, 1991

Az alábbi üzenet csak a vírus visszafejtése során válik láthatóvá:

Non rompetemi le palle o mi arrabbio...

non so se sono stato abbastanza chiaro...

A vírus a csak olvasható (read only), a rejtett (hidden) rendszerállományokat, a normál programokat és a lemez címkéjét (label) is megfertőzi. A fertő-

Vírusíró automata

Már korábban sejtettük, hogy a vírusírást is automatizálták. A nyugatnémet számítógépes alvilágban használt új szoftvert végre sikerült megszerezni. Ez nem más, mint egy víruskészítő automata, amely a programozásban járatlanok számára is lehetővé teszi vírusok írását.

A program képernyőfotóiból is kiderül, hogy kulturáltan megírt szoftverrel van dolgunk. Elindítása után bekéri a nevét egy szövegállománynak, amelyben elhelyeztük a vírusba rejtendő üzenetet. Utána a következő lépésben megkérdezi, milyen generációs mutációt válasszunk. Ennek értéke 1 és 100 között lehet. Amikor a program mindezt megkapta, generál egy 1074 bájt hosszú állományt, amely a generikus, azaz ősvírus. Ha ezt elindítjuk valamelyik gépben, akkor innen épül bele más állományokba, hogy elkezdje terjedését.

A Virus Construction Set v.1.0 egy bináris állományt és egy igen rövid, eddig általunk ismeretlen tömörítő eljárással többszörösen összenyomott programállományból áll. A bináris állomány tartalmazza az ősvírus bináris kódját, amelyet a szoftver tovább módosít. Mi egy ilyen könyvtárat tudtunk megszerezni, de valószínűleg már több is létezik belőle. A rendszer által generált vírus visszafejtése most van folyamatban.

A program gyaníthatóan a Whale és a Fish-6 írásáról hírhedtté vált hamburgi vírusfejlesztő csoporttól való. (Állítólag már több ilyen vírusfejlesztő készlet is került forgalomba Európában.)

Sajnos nemcsak fejlesztőkészlet került így forgalomba, hanem új vírustechnológia is született. Elkészítették az ANSI-bombát, amely nem normál programkód, hanem az ANSI.SYS kódkészletét felhasználva ANSI szekvenciaként programok képernyőrutinjaiba építve végez pusztítást gépünkben. Kimutatására a hagyományos eljárások alkalmatlanok.

```
                 V i r u s   C o n s t r u c t i o n   S e t
              (c) 1991 by VDV - Verband Deutscher Virenliebhaber

VDV international präsentiert: Virus Construction Set V1.0

Für einen Virus benötigen Sie ein Textfile von maximal 512 Bytes Länge.
Dieser Text wird dann nach einer vorgegebenen Anzahl von Generationen aus-
gegeben. Außerdem werden AUTOEXEC.BAT und CONFIG.SYS gelöscht. Ansonsten
wird kein Schaden angerichtet, die größeren Gemeinheiten hebe ich mir für
meine eigenen Viren auf (hehe...)

Dateiname der Text-Datei ?

Es wurde jetzt eine Datei VIRUS.COM erstellt, die Ihren Virus enthält.
Um ein Programm (nur .COM-Dateien) zu infizieren, gehen Sie wie folgt vor:

1) Kopieren Sie das zu infizierende Programm auf eine leere Diskette:
   C:\>copy programm.com a:
2) Wechseln Sie auf das Diskettenlaufwerk:
   C:\>a:
3) Starten Sie VIRUS.COM:
   A:\>c:virus
Das infizierte Programm ist dann 1077 Bytes länger als vorher und kann
an Bekannte, Freunde oder Feinde weitergegeben werden.

Taste drücken, um zum DOS zurückzukehren
```

zött állomány mérete nem változik meg, csak akkor, ha a fertőzött program rövidebb volt, mint a vírus hossza. Ha a vírus fertőzése során hiba lép fel, és rendszerdátuma hétfőt mutat, akkor a C: lemez első 160 szektorát tönkreteszi. A merevlemez legyalulását egymás után 160-szor ismétli meg. Ha munkájával végzett, akkor a következő üzenetet jeleníti meg:

Exterminator Virus 1.0 (c) by Cracker Jack 1991 (IVRL)

No panic...this is a Harmless Virus...

A vírus ismert mutánsai:

A mutáns hossza 256 bájt, viselkedése megegyezik a fent leírt módszerrel, azzal a különbséggel, hogy hétfőn nem polírozza a harddiszket, hanem a CGA/EGA monitor vízszintes szinkronjelét tologatja jobbra-balra. A vírusban a következő szöveg található:

BadGuy Virus (c) by Cracker Jack 1991 (IVRL)'

Italian Virus Research Laboratory (C) 1990, 1991

IVRL Head Quarter, Milan Italy

Demon

A Demon egy víruscsaládba tartozik az alábbi copyrightot viselő vírusokkal, azonos kódoló rutint tartalmaz, s ugyanannak az olasz üzemnek a terméke. Üzenete:

„Demonhyak Viri X.X (c) by Cracker Jack 1991 (IVRL)"

„Error eating drive C:"

Label

Az újabb vírusok megfertőznek bármilyen attribútumú állományt, beleértve a lemez címkéjét is. Az Exterminator és a Label is ilyen. A lemez címkéje gyakorlatilag fájlként kezelhető. Némi furfanggal írni lehet bele, és olvasni lehet onnan. Ez ad lehetőséget a lemezcímkék megfertőzésére.

A Sysdoki installálása során szintén a merevlemez címkéjének paramétereit tárolja el, és ennek megfelelően működik demó- vagy teljes verzióként. Ez az oka annak, hogy ha a szoftvert lemásolják a merevlemezről és átviszik egy másik gépre, ott az csak demóként működik. Az Exterminator és a Label vírus a Sysdoki jogos felhasználói példányából szintén demóverziót készít. Ilyen típusú vírusok jelenléte is magyarázat lehet arra, ha egy felhasználó a Sysdokit csak demóként tudja használni. (A Sysdoki floppyról csak vírust keres, irtani a merevlemezről, fájlból tud — memóriából nem.)

Thunderbyte 1.x

Vírusőrjárat rovatunkban korábban már szó volt róla, hogy a Thunderbyte vírusvédelmi rendszerre könnyen lehet olyan vírust írni, amely simán átmegy a kártyán. Ez nem is nehéz feladat, még a Polimer vírus is elmegy a Thunderbyte mellett. A Thunderbyte vírus szerzője — valószínűleg az olasz vírusírók iskolájának neveltje, a Plague vírust vette alapul, és azt barkácsolta át „kártyajáróvá". A vírusban lévő szöveg:

Metal Thunder Virus — Ver. 1.X

(C) by Metal Thunder

-* IVRL *- All rights reserved

Ez a vírus BIOS-szintű sávformázást végez, ami végleges adatvesztést jelent a felhasználónak.

FellowShip

Már régebben, egy nyugatnémet szoftverszállítmánnyal együtt érkezett Magyarországra a FellowShip vírus. Akkor az importőr — félve a következményektől, az esetleges perektől — nem volt hajlandó a köz érdekében a vírust rendelkezésre bocsátani. Időközben a vírust egy felhasználó eljuttatta hozzánk. Érdekességképpen a vírusban lévő szöveg:

This message is dedicated to all fellow PC users on Earth

Towards A Better Tomorrow

And A Better To Live In

A „rendőrgyilkos"

Nem megerősített információk alapján Magyarországon kering egy vírus, amely legelőször a víruskereső és vírusölő programokat — tehát a rend őreit — támadja meg. A hír szerint a vírus a rendszerbe jutva megkeresi a víruskereső és vírusölő programokat (SCAN, PCSCAN, CLEAN, TNTVIR, SYSDOKI stb.), majd törli azokat a lemezről. Amint ez megtörtént, bejelenti, hogy átvette a vezérlést, és a gép teljesen az övé. A híreket Szeged és Debrecen térségéből kaptuk, valószínűleg az egyik ottani vírusíró egyetemista programozó tesztelés alatti termékéről van szó. (Ha valaki találkozik a vírussal, küldje el!)

Tömör gyönyör

A tömörített állományú vírusok létrehozásának lehetőségét több mint másfél éve ismerjük, de még csak most jutott el hozzánk az első ilyen példány. A recept nagyon egyszerűnek látszik. Végy egy nem fertőzött programot. Fertőzd meg valamilyen vírussal, majd nyomd össze olyan állománytömörítővel, amely futás közben csomagolja ki magát (PKLITE, LZ, SCRNCH stb.). Az összezsugorított vírust a normál víruskereső programok nem ismerik fel, mert ahhoz ismerniük kellene a tömö-

INFORMÁCIÓKÉRÉS: 29 ▶

Megjelent az új TNT!

A Carmel izraeli szoftverfejlesztő cég hazánkban is ismert TNT antivírus programjának másolásvédelem nélküli új változata most került be a hazai szoftvercsere csatornáiba. A program verziószáma 7.06A, parancsnyelve német.

A programhoz tartozik a Defender nevű kis tárrezidens vírusvédelmi program, amely McAfee VSHIELD programjához hasonlóan viselkedik, vírusismerete alapján próbálja megakadályozni a vírusok memóriaműveleteit. Javított változat a BOOTSAVE segédprogram is.

Maga a főprogram hálózati környezetben is jól működik. Sajnos a gyártó cég nem tudott letenni az immunizálás rögeszméjéről. Így a SAUBERN+IMMUNIZE menüpontot programjaink érdekében soha ne használjuk! Mindig csak a keresés (Suchen) vagy pedig a kitisztítás (Saubern) funkciót alkalmazzuk.

A program most már nem olyan rossz, mint korábbi változata (amit az 1990-ben megjelent Víruslélektan könyv részletesen bemutatott), és ezzel a verzióval az elfogadható kategóriába küzdötte fel magát. Kijavították a Yankee Doodle és a Vacsina vírusismereti hiányainak nagy részét, s már nem minden esetben teszi tönkre az állományokat.

Sajnos a Norton Antivirus Toolkithez hasonlóan első látásra nem lehet (a Nortonnál azután sem!) eldönteni, hogy irtja vagy csak keresi az adott vírust. Ne lepődjünk meg, amikor kitesz egy táblát, hogy új vírussal áll szemben, és azt küldjük meg a forgalmazó címére, s akkor ő majd küld nekünk egy új programot, cserébe a fáradozásért. A magyar Kukac vírust is felismeri, de nem irtja ki. (Nem éri meg beküldeni, mi már megtettük...) Mindenesetre nem erősségük a magyar nyelv.

Sajnos a magyar bennszülött vírusok egy részét továbbra sem ismeri fel. Ugyancsak elsiklik a DOS62 vírus felett. Másodlagos víruskeresőnek és irtóprogramnak ajánlható, de az immunizálás funkció használata továbbra is „életveszélyes". Mindenesetre látványos, amikor dolgozik.

rítő algoritmust, amelyet a szoftvergyártó cégek nem adnak ki. Egy-két algoritmust azért beépítenek a víruskeresőkbe. Például a SCAN ismeri az LZEXE algoritmusát, a CHKVIR az LZEXE és az EXEPACK-kal tömörített állományt. A PCSCAN szoftverben is több ilyen visszafejtett algoritmus van. A PKLITE teljes verziója, amit egyre többen használnak — az anti-debug változat — azonban igen komoly lehetőség a vírusok elbújtatására.

Ha ezt a megoldást több vírussal megismétlik, és mindig más tömörítő algoritmust használnak, akkor könnyen elkészíthető egy „tökéletes" vírusszendvics.

A víruskereső programok soha nem fogják tudni, melyik program a hordozó, legfeljebb a másodlagosan fertőzött állományokból tudják kiölni a vírust, ha az nem az új generációhoz tartozó, felülíró típusú. Az ilyen módon elkészített vírusos állományt csak rezidens víruskereső programmal vagy valós idejű vírusellenőrző programmal lehet detektálni.

Felmerül a kérdés: hány vírus felismerésére alkalmas rezidens víruskereső programot ültessek be rezidensen, hogy biztos legyek a dolgomban?

CRC ellenőrzés és fájlimmunizálás

A TG vírustesztje során Norton Commandert használva kiderült, hogy a „forgalomban" lévő vírusok közül milyen sok támadja meg a COMMAND.COM programot. A másik tapasztalat az, hogy a Norton Commander NCMAIN.EXE programját akarják elsőként megfertőzni. Ez érthető is, mivel a rezidens NC.EXE mindig ezt a programot hívogatja. Az ilyen felhasználói környezetben a fenti fájlokra külön figyelmet kell fordítani. A tapasztalat szerint az NC.EXE tűri a Buruzs Tamás-féle SPS rendszer ráültetett önvédelmi rutinját. Így legalább az visít, ha vírus került a rendszerbe.

Mennyire hatékonyak a CRC ellenőrző programok és a fájlimmunizáló (SPS, SYSDOKI) kis rutinok? Ezek a programozástechnikai megoldások a vírusvédelem korábbi stádiumában elégségesek voltak, és hatékony megoldásokat adtak. Ma már azonban olyan vírusok vannak, amelyek ellen ezek a módszerek hatástalanok.

Az elkövetkező időszak fő vírusirtási metódusa a fertőzött fájl törlése lesz. Már most nagyon sok vírus eltávolításánál olvashatjuk a „delete infected file" üzenetet. Ehhez is szükség van azonban arra, hogy egyáltalán legyen mit törölni, és legyen honnan törölni, amire a Patricia, Mirror stb. vírusok esetében nem sok esélyünk marad.

A CRC ellenőrző eljáráson alapuló, változásmérő detektorok egy ideig hatásosak voltak, de ezek csak annyit tudnak megállapítani, hogy a fájl tartalma megváltozott — már amennyiben a vírus megengedi nekik, hogy az eredeti fájltartalmat lássák, mert a 4096-oshoz hasonló vírusok nem ilyen kegyesek. Ez tehát nem egy biztos megoldás. Ráadásul a CRC ellenőrző programokkal csak a tartósan változatlan fájlokat célszerű ellenőrizni, a gyakran változó tartalmúakra alkalmazva igen kényelmetlenek.

A dBase vírus az egyik olyan vírus, amely ezt az elvet figyelembe veszi. Mindamellett, hogy a vírus terjed, a felhasználó DBF állományát kódolva létrehoz egy olyan állományt is, ahol a dekódoló algoritmust tárolja. Amikor már a teljes adatbázist kódolta, akkor a DBF állományt egyszerűen kitörli.

Körülbelül két évvel ezelőtt még azt hittem, hogy a fájlimmunizáló programok hatásos általános célú vírusvédelmet tudnak nyújtani. A jelenlegi vírushelyzet megváltoztatta a véleményemet. Ha a vírus újraformázza a merevlemezt, akkor igazán mindegy, hogy voltak-e a fájlok immunizálva vagy sem. Ha a vírus felülírja (tönkreteszi) az állományokat, akkor rá sem fut az immunrutinra, de nem is lenne mit helyreállítani (1, 2, 8 kbájt programrészletek). Sajnos fel kell adni az ilyen jellegű fejlesztést is. Kellemetlen érzés belátni, hogy az az út, amelyen eddig jártam, tovább már nem vezet sehova. Vissza kellett térnem a kiindulási pontra, és elindulni egy egészen más, eddig még járhatatlannak bizonyult úton.

A felhasználók szempontjából ez látszólag visszalépést jelent a vírusölő programok tudási szintjében. Ha viszont a vírus működésének eredményeképpen az állományok tönkremennek, akkor teljesen mindegy, hogy milyen vírusölő programot használunk, egyedül az a fontos, hogy minden fertőzést megtaláljunk. Ha detektáltuk, hogy melyik állományban van a vírus, akkor a legjobb megoldás felülírni az eredeti szoftverrel.

Szegedi Imre

Szovjet vírusinvázió

1990 végén a számítástechnikai szakma egyik kellemetlen meglepetése volt, hogy robbanásszerűen megjelentek a szovjet vírusprogramozók remekei. Magyarország valószínűleg nem közvetlenül, hanem Ausztria és Csehszlovákia felől kapja majd egyre nagyobb mennyiségben ezt az „áldást". Az elkövetkező időszakban könnyen lehetünk mi is a szovjetek által kifejlesztett új terjedési algoritmusok, hordozórutinok gigantikus béta-tesztjének tanúi — és szenvedő alanyai.

A különböző csatornákon kapott hírek szerint a Szovjetunióban több nagy vírusgyártó góc is létezik. Katonai fejlesztések folynak Kijevben és Szentpétervárott. Moszkvában az Informatikai Intézet és egy-két egyetem lehet a fő víruskibocsátó. Viszonylag kevés a magányos farkas, de ez elsősorban a magánszemélyek rossz gépellátásának tulajdonítható.

Az alábbiakban két vírust bővebben ismertetünk, a többit a mellékelt táblázatban soroljuk fel a fertőzés típusa és a vírushossz feltüntetésével. (Mindegyikről részletes leírást és dokumentációt közöl a Compfair '91 idejére megjelenő Új víruslélektan c. könyv, az Alaplap Könyvek sorozatban.)

V-311

Ez a szovjet vírusfejlesztés új irányzatának egyik „örömprogramozott" darabja. Írója beletette mindazt az ötletet, amelyet tanulmányai során merített. A cél látható volt, minél kisebb méretbe minél több funkciót bepakolni. A vírus 1991 januárjában bukkant fel Nyugat-Európában, majd az USA-ban.

A vírus nem rezidens, .COM-fertőző, parazita program, hossza 311 bájt. Azonosítani lehet a Scan v74+ programmal, eltávolítja a SCAN /D parancs, illetve a fertőzött állomány törlése.

A program csak a .COM állományokat fertőzi meg, nincs rezidens része. A COMMAND.COM-mal manipulál, ugyanis ha a fertőzött program kódja lefut, a programvírus kinéz a rendszerórára. A vírusban 16, előre meghatározott érték van. Ha a lefutás során az óra ezeknek a beállított értékeknek egyikét mutatja, akkor a vírus a COMMAND.COM-ot átnevezi COMMAND.CON-ra. Ennek hatására a gép az ilyenkor szokásos „Cannot load COMMAND.COM, System halted" rendszerüzenettel lefagy, illetve ha rendszert akarunk indítani, az istennek sem indul el. Ha az óra nem egyezik egyik előre beprogramozott lehetőséggel sem, akkor a vírus az aktuális könyvtárban megfertőz egyetlen .COM állományt, és nem nevez át semmit sem.

A megfertőzött program 311 bájttal lesz hosszabb, a víruskód az állomány végére épül be. Amelyik alkönyvtárban már fertőzött, ott megtartja ugyan az állományok eredeti dátumait, de megváltoztatja az alkönyvtár létrehozásának a dátumát a következőre: 11:19:32. Ez a fertőzés jele a vírus számára. Ez azonban nem minden! Átállítja az egyes állományok attribútumbitjét az alkönyvtáron belül, mégpedig ha 8-15 között van, akkor reseteli. Ezzel néhány ezt alkalmazó program (például a backup) meglepetésszerű dolgokat művel.

Akuku

1991 januárjában egy szovjet vírusműhely újabb szellemességgel lepte meg a világot. A benne lévő üzenet alapján kapta az Akuku nevet. Minden fertőzött programban megtalálhatjuk a következő, épületes orosz nyelvű üzenetet, latin betűkkel írva:

A kuku, Nastepny komornik !!!

A vírus parazita, nem rezidens. A fertőzés a .COM és az .EXE állományokra egyaránt kiterjed. De ami nem erény a vírusvilágban: a COMMAND.COM-ot is megtámadja. A programozó valószínűleg nem lehetett teljesen tisztában az .EXE állományok szerkezetével, mert fertőzés után az .EXE állományok sok esetben nem működnek, és az „Error in EXE file" rendszerüzenettel örvendeztetnek meg bennünket.

A vírus áldozataira meglepetésszerűen csap le az egyes meghajtókon. Amikor a víruskóddal fertőzött program lefut, az aktuális könyvtárban három programot megfertőz. Ha már nem talál fertőzetlen programot, akkor kezd el keresni egyéb helyeken, például a C: meghajtó gyökérkönyvtárában. A fertőzés hossza 891-907 bájt között változik, és a kód a fertőzött program végére épül be. Az eredeti állománydátumot és időpontot nem bántja, azok a fertőzés után is megmaradnak eredetinek.

A Scan v 74+ felderíti, eltávolításához legjobb törölni a fertőzött állományt.

Áttekintés a szovjet vírusújdonságokról, méretük szerinti növekvő sorrendben. Név (zárójelben egyéb elnevezése); milyen állományokat fertőz meg; szokásos hossza bájtban, ha a név nem, vagy nem ugyanazt tartalmazza:

V-311 (USSR 311) — .COM
USSR 492 — .COM, 495-508
Leapfrog (USSR 516) — .COM
USSR 576 — .EXE, 576-586
USSR 707 — .COM
USSR 711 — .COM, 705-717
Akuku (USSR 891) — .COM, .EXE, 891-907
USSR 948 — .COM, .EXE
Bebe — .COM, 1004
USSR 1049 — .COM, .EXE, 1051-1064
Voronezh (Voronyezs) — .COM, .EXE, 1600
Russian stealth (USSR 1689, SVC V4.00) — .COM, .EXE
USSR 2144 — .COM, .EXE, 2144-2159
Voronezh related — .COM, 2200

DECnet programféreg

A vírus belép a hálózatba

Az Alaplap 1991. áprilisi számában közöltünk egy DEC számítógépre írt vírusrészletet. Nem ismervén jól a DEC rendszerkörnyezetet, kértük az ezzel foglalkozó szakembereket, hogy segítsenek megfejteni a különös szövegállomány-vírus rejtélyét. A felhívásra a Magyar Távközlési Vállalat Távközlési Informatikai Központjában üzemelő országos kísérleti DECnet hálózat két rendszergazdája, Cser László és Balázs Béla jelentkezett. Segítettek a vírus visszafejtésében, ugyanakkor felajánlották segítségüket a DEC gépek vírusproblémáinak megoldásában. (Telefonjuk: 157-0570/101.) Az alábbiakban közöljük nyomozási beszámolójukat.

Amikor elolvastuk az Alaplapban megjelent cikket, kissé idegesek lettünk. Egy országos, nagy megbízhatóságú VAX/DECnet hálózat rendszergazdáiként úgy tudtuk, hogy a Digital Equipment Corporation többszintű hozzáférésvédelmi-önvédelmi rendszerét eddig senkinek nem sikerült kijátszania. Mielőtt az e témával foglalkozó többi kolléga is ideges lenne, gyorsan eláruljuk: ez a vírus sem képes ilyesmire, és bizonyos óvintézkedések betartása esetén nem is életképes.

Az állomány .COM kiterjesztésű. De nem mind arany, ami fénylik, azaz nem minden fut a PC-n, ami .COM állomány. Mivel ez szövegfájl, akkor lenne futtatható a PC-n, ha .BAT állomány lenne, amely a DOS számára értelmezhető parancsokat tartalmaz. Nos, a cikkben szereplő programrészlet is egy parancsállomány, de nem a megszokott PC-s DOS környezet számára.

A cikk feltételezése helyes volt: a program VAX/VMS környezetből való. Mégpedig mindez a VAX/VMS operációs rendszer kommunikációs nyelvén, DCL-ben íródott (DCL = Digital Command Language). Ebben a környezetben a .COM kiterjesztés nem bináris végrehajtható állományra, hanem a command file (parancsfájl) elnevezésre utal. A parancsfájl hasonló funkciót tölt be, mint PC-s környezetben a .BAT batch-fájl. A felhívásban megjelent részlet alapján, első ránézésre ennyi volt megállapítható a „DECnet vírusról".

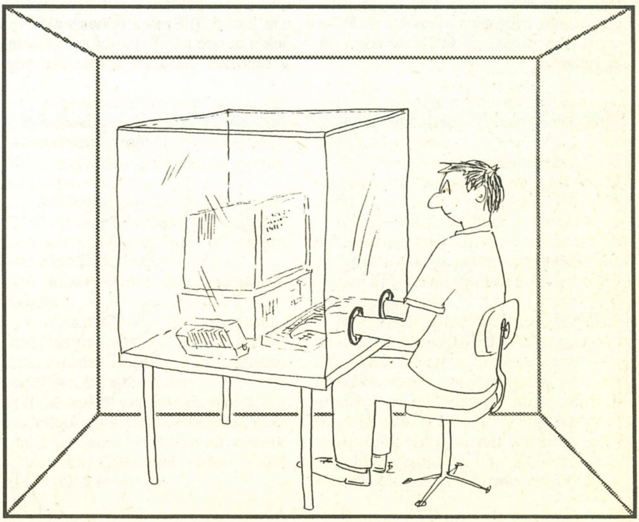
Jelentkezésünk után megkaptuk a parancsfájl teljes listáját. Rövid analizálás után megállapíthattuk, hogy helyesen tették, amikor nem közölték az egészet, mert több érdekes trükk van benne, amelyek bizonyos rendszerhiányosságokat használnak ki. A teljes lista alapján vissza lehetett fejteni a programot, amely lényegében vírus.

Olyan értelemben vírus, hogy terjed a rendszerben. Viszont jóindulatú, mert terjeszkedése során önmagán kívül mást nem töröl, illetve nem épül be más programokba.

Ha a klasszikus vírusosztályozást vennénk figyelembe, akkor a programférgeknek (angolul worms) a vírusokénál kevésbé népes, de sok elegáns programozási fogást tartalmazó csoportjába tartozik.

Miben nyilvánul meg mindez?

— Nem fertőz, mert más parancsfájlokhoz (.COM) vagy végrehajtható programokhoz (.EXE) nem nyúl. Nem is nagyon tudna hozzájuk nyúlni, mivel a VAX/VMS többfelhasználós operációs rendszer, és emiatt jól ellátták hozzáférésvédelmi rendszerrel.



Többszörös vírusvédelem

— Nem okoz kárt, mert nem töröl (önmagán kívül) semmit, nem állítja meg a gépet (csak alaposan lelassíthatja), és nem is formattálja a lemezeket. A VAX/VMS alatt nem is lehet formattálni. Bosszantásra csak karácsonyi üdvözleteket küld.

— Terjeszkedik, de csak abban az esetben, ha az a VAX gép, amelyen elindították, része egy számítógép-hálózatnak, és a hálózat DECnet-alapú. Adathordozó (hajlékony mágneslemez) közvetítésével nem terjed.

Természetesen a fenti megállapításokkal vitába lehet szállni. Például joggal kérdezhetik: Mi az, hogy nem fertőz? Hiszen fertőz, ha a számítógép-hálózatot tekintjük egységes rendszernek, valamiféle „hipergép"-nek, mert ez a program a hálózat csomópontjai között vándorol, s ott többszörözi magát!

Ennyi bevezető elméleti eszmefuttatást szükségesnek éreztünk elmondani a nagygépes rendszerek lelkéről. S most nézzük meg részletesebben, mit is csinál valójában ez a programféreg!

Ezt a programot kizárólag azért készítette valaki, hogy segítsen szétküldözgetni a karácsonyi üdvözleteket. Ugyanis a program csak 1988. december 24-én nulla órától kezdve 30 percen át fejtette ki hatását.

Természetesen, ha az időkorlátozást kivesszük belőle, akkor bármikor bármeddig képes működni, de úgy nagyon könnyen lefülelhető. Néhány jellemző tevékenységének ismertetése során ne lepődjünk meg, hogy sok olyan dolgot csinál, mint egy megszokott PC-s vírus. Az elvek, a működés logikája ugyanaz!

— A program a saját, futás közbeni azonosítóját MAIL_178DC névre változtatja. Ez neki több okból is „hasznos": álcázza magát, valamint fel tudja venni a kapcsolatot a VAX/VMS MAIL segédprogramjával, amely levelezési lehetőséget biztosít a hálózaton dolgozó felhasználók között.

— Beolvassa önmagát a memóriába, majd a lemezről kitörli magát (delete hi.com;*). Ugye, ezt egy PC-s vírus is megtehetné!

— Generál egy hálózati csomópont-azonosítót, kvázi véletlenszerűen, felhasználva az aktuális időt. Ez a csomópont lesz az, ahová átmásolja, majd ott elindítja önmagát. Azt természetesen ellenőrzi, hogy a hálózatban van-e olyan azonosítójú csomópont, amilyet előállított. A másoláshoz és indításhoz felhasználja azt, hogy a DECnet hálózatban lévő VAX gépeken általában definiálva van egy alapértelmezés szerinti hálózati bejelentkezési lehetőség. (A VAX/VMS rendszerbe csak felhasználói név és jelszó ismeretében lehet belépni.) Ez az alapértelmezés szerinti belépési lehetőség nem rendelkezik privilégiumokkal.

— Az általunk elemzett változatnak 1988. december 24-én nulla óráig volt lehetősége terjeszkedésre. A program szerint amint elérkezik ez az időpont, hozzáfog az üdvözletek szétküldéséhez. Nagyon ügyes fogással megszerzi majdnem az összes felhasználó nevét azon a csomóponton, amelyiken működik. (Nos, például ezért nem publikus a teljes kód!) Ezek a felhasználók lesznek a címzettek. Felveszi a kapcsolatot a rendszer hálózati levelezést végző programjával, és a parancsfájlban lévő üdvözlet szövegét elküldi a kiválasztott címzetteknek a Télapó nevében.

— Kitörli azt a fájlt, amelybe a megszerzett felhasználói neveket tette, majd leállítja önmagát (STOP/ID=0).

Ezután joggal vetődik fel, hogy mit lehet tenni egy ilyen trükk elkerülésére? Mi az alábbi következtetéseket vontuk le, és azokat ajánljuk más rendszerek működtetőinek is:

— Szigorúan be kell tartani a VAX/VMS védelmi előírásait.

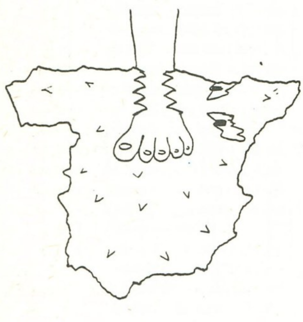
— A hálózat gépei semmiképpen ne legyenek átjárhatók az alapértelmezés szerinti (éppen ezért közismert) bejelentkezések segítségével. Minden bejelentkezési lehetőség egy-egy konkrét személyhez kapcsolódjon.

— A rendszer felhasználói mindig csak a munkájukhoz feltétlenül szükséges minimális privilégiumokat kapják meg.

— A SYS$SYSTEM:RIGHTSLIST.DAT fájlhoz ne legyen bárkinek még olvasási joga sem. Ha ehhez hozzáfér valaki, akkor a kulcs egy része már kezében van a rendszer manipulálásához.

A probléma szerencsére még idejében nyilvánosság elé került. Reméljük, hogy a nagygépes hálózatokon nem kell komolyabb vírusinvázióval számolni, és nem bukkannak fel az ARPANET amerikai félkatonai rendszer leállítását okozó behatolás epigonjai sem. Ehhez viszont a rendszerüzemeltetőknek és a felhasználóknak komolyan kell venniük az adatbiztonsági szabályokat.

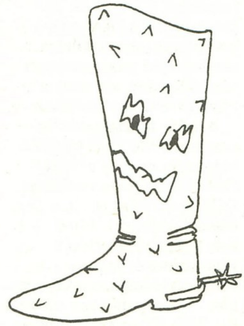
Cser László — Balázs Béla



Spanyol boot-vírus



Olasz boot-vírus



Magyar boot-vírus

Top Guard

Víruskártya-játszma

A szoftveres vírusvédelem problémáit látva külföldön már korábban elkezdtek intenzíven foglalkozni a hardverben rejlő lehetőségekkel, és piacra dobták az első hardveres vírusvédelmeket. Először mi is a kereskedelmi forgalomban lévő vírusvédelmi kártyák között kerestünk olyanokat, amelyek megfelelnek az általunk támasztott követelményeknek. (Lásd a mellékelt keretes anyagban lévő 14 pontot.) A kártyák tesztelése során szerzett rossz tapasztalatok azonban egyre inkább saját vírusvédelmi kártyánk megtervezésére, kifejlesztésére ösztönöztek bennünket, s e munka eredményeként megszületett a Top Guard, amelynek bétaváltozata az idei budapesti Ifabón lépett a nyilvánosság elé.

A fejlesztés során az a vélemény erősödött meg bennünk, hogy a jövő vírusvédelmi eszköze olyan hardveres megoldás lesz, amelyet szoftver támogat. A koncepció gyakorlati kivitelezésének kulcsa — és az eddigi kártyák leggyengébb pontja — a vírusok ismerete. Általános ismérvekkel a jelek szerint aligha lehet minden jelenlegi és jövőbeni vírust meghatározni.

A másik fontos alapelv volt, hogy a kártyában rejlő lehetőségek kiaknázására és a védelem továbbfejlesztésére ne kelljen később mindig új kártyákat készíteni (a felhasználónak pedig megvenni), hanem legyen mód a program továbbfejlesztésére, illetve a már meglévő egységhez kiegészítők, „kártyaoldalkocsik" kapcsolására. Jól látható fejlődési tendencia továbbá a számítástechnikai rendszerekben tárolt adatok és programok sérülés és lopás elleni biztosítása. (Amit röviden vírusbiztosításnak nevezhetünk.) A kártyát ezért kiegészítő szoftverekkel és elektronikával bizonyos adatvédelmi és vagyonvédelmi funkciók ellátására is alkalmassá kellett tenni.

Hogyan is működik a TG kártya? A fejlesztés a vírusvédelmet két fő szakterületre osztotta fel:

1.) Ismert vírusok elleni, konkrét vírusismereten alapuló védekezés.

2.) Ismeretlen vírusok elleni, általános eljárásokon és algoritmusokon ala-

A jó vírusvédő kártya követelményei

1. Lehetőleg minél kevesebb döntést bízzon a felhasználóra.

2. Csak akkor „szóljon", ha valóban baj van és nem tud mit csinálni. Különben maradjon teljesen észrevétlen.

3. Használata ne függjön a felhasználó akaratától, tehát ha bent van, akkor valóban védjen.

4. Rendelkezzék megfelelő vírusismerettel. A „katalógusvírusokkal" történő fertőzéseket minden körülmények között akadályozza meg.

5. Legyen a vírusfunkciók felismerésére alkalmas általános védelme. Ne engedje meg az érzékeny rendszerállományok és részletek (CMOS-RAM, boot, partíciós tábla, rendszerállományok) felülírását, a merevlemez szektorainak formázását.

6. A processzorok minden üzemmódjában úgy működjön, hogy védő hatása érvényesüljön. Például az MS-Windows védett (protected) módját is ismerje.

7. Ne foglaljon el helyet a memóriából, és már a rendszer betöltődése után, a merevlemez boot folyamata előtt aktivizálódjon.

8. Minden winchestertípuson működjön.

9. A vírusok által fizikailag megtámadhatalan legyen, azaz a kártya lássa a memóriát, de a memóriából ne legyen látható a kártya.

10. A kulcsparaméterek tárolására rendelkezzen saját memóriával, akár onnan is lehessen folytatni a boot folyamatot. (Elemmel védett RAM vagy ROM is lehet.)

11. A vírusokra vonatkozó ismeretanyaga könnyen felfrissíthető és bővíthető legyen.

12. Támogassa a hozzáférésvédelmet és az adatbiztonsági szolgáltatásokat.

13. Semmilyen általánosan elterjedt szoftver (például Norton Utilities, Norton Commander, Ventura stb.) futását ne akadályozza és ne is nehezítse meg.)

14. Felhasználói programokból rendszerhívásokkal elérhető legyen, támogassa a programozókat. Rendelkezzék olyan programinterfésszel, amely más alkalmazási szoftverből meghívható. (Például ha egy könyvelőprogramhoz menet közben be kell hívni egy floppyt, akkor tudja ellenőrizni annak vírusmentességét.)

puló védelem, amely a vírusfunkciót felismeri és leblokkolja.

Az ismert vírusok elleni védekezés az eddigi szoftveres eredmények felhasználására épülhetett. A kártya szoftvere a BIOS után, de még az operációs rendszer betöltése előtt aktivizálódik, operációs rendszertől független fájlkezeléssel. Itt még egy problémával találkozott a programfejlesztő, ami miatt a szoftverírók általában elkerülik az ilyen megoldásokat: igen sokféle merevlemeztípus és kontroller van forgalomban, és a védelemhez használandó, nem publikált vezérlőparancsok jelentősen eltérnek. Hasonló gondokat okoznak a speciális alaplapok is.

Ha a kártya ismeretlen vírussal találkozik, akkor az általános célú védekező rutin működik. De egy ismert vírussal találkozva is csak a vírusra konkrétan nem utaló, általános üzenetet tud kiírni. („Interruptcsere. Engedélyezi? Igen/Nem", „A fizikai #1 lemezcilinder, 0. fej, 1. szektorába ír a program. Engedélyezi? Igen/Nem".) Ilyen hibaüzenetek az általános célú vírusvédelemnél másodpercenként fellépnek, s azokat csak akkor kell közölnünk a felhasználóval, ha már a mintegy 200-400 vírusra és annak változataira ellenőriztük a programot, mégis van valami rendellenesség.

A vírusvédelmi kártyának a DOS operációs rendszer betöltődése előtt kell elvégeznie a boot-vírusok ellenőrzését. A TG kártya ismert boot-vírusok jelenléte esetén kiírja a vírus nevét, és igény szerint ki is takarítja őt a floppylemezről.

Ha ismeretlen boot-vírussal találkozik, akkor az általános védelmi rutin lép működésbe. Ez a védelemi mechanizmus tájékoztatja a felhasználót, és nem engedi aktivizálódni a vírust.

Ha valamelyik boot-vírus mégis kijátszaná a kártya hardver- és szoftvervédelmét, megfertőzve a merevlemez boot-szektorát, a vírusvédelmi rendszer azt is észleli, és helyreállítja az eredeti információkat az akkumulátorral védett memóriában tárolt eredeti boot és partíciós tábla információi alapján.

A fájlvírusok elleni védelmi rendszer a következőképpen müködik: minden program indítása előtt ellenőrzi az összes programot az ismert vírusokra. Ez a művelet olyan gyorsan történik, hogy a felhasználó észre sem veszi. Amennyiben a rendszer ismert vírust talál, akkor azt el is távolítja, és a műveletet automatikusan dokumentálja.

Nem mellékes szempont, hogy a védelmi programok ne foglaljanak el helyet a gép memóriájából. Ezt azzal lehetett elérni, hogy egy sajátos technika segítségével a program nem a gép főmemóriájában fut, hanem a kártyán lévő, 256 kilobájtos RAM-ban.

Amikor aktivizálódik az általános célú vírusvédelmi rendszer, az megakadályozza az ismeretlen vírusok szétterjedését és fertőzését. Ha valamilyen vírusgyanús műveletnél a felhasználónak kell döntenie, természetesen arról is dokumentum készül. Mindezeken túl a TG kártyával megoldható a floppylemezegység letiltása, szükség esetén csak a rendszergazda engedélyezheti új programok felvitelét a merevlemezre, vagy csak érvényes jelszó megadása esetén válik lehetővé a gép használata.

Ha megtörtént a baj, és a felhasználó számítógépes rendszerébe bekerült a vírus, nem a legelső szempont annak sietős kitakarítása. Sokkal fontosabb, hogy a helyreállított állományok jók legyenek. Ez egyetlen szoftverféleségnek nem kedvez: amelyet üldözési mániás programozók írtak, mert az a legkisebb vírus-piszkálgatásra is üt, mind a bolondóra...

A TG vírusvédelmi rendszer fel van készítve olyan vírusok ellen is, amelyek a „Ctrl-Alt-Del" újraindítást kibírják. Ha a kártyát egy CMOS-szal (AT/286/386/486) rendelkező számítógépbe helyezzük, a védelem megakadályozza a CMOS-ba író programok és másolásvédelmek (például a Lotus 1-2-3 v3.1 magyar változat) futását, de cserébe védelmet nyújt minden olyan vírus, trójai program vagy büntető másolásvédelem ellen, amely a merevlemez típusának menet közbeni átírásával teszi tönkre a merevlemezt.

A TG koncepciója szerint installálása után a védelmi rendszer csak kis mértékben függ az embertől. Nem kell rendszeres vírusellenőrzést tartani, mivel az állandóan folyik. Nem kell semmilyen vírusölő programot lefuttatni, mert minden program ellenőrzésre került — még a számítógépbe való bejutása előtt (real time módban, észrevétlenül).

A szokásos szoftveres megoldásokkal ellentétben a kártya teljes körű naplózási rendszere pontosan kimutatja, hogy hogy melyik program volt a vírushordozó, és mikor került a rendszerbe. Ez részletkérdésnek tűnik, pedig igen fontos kideríteni a fertőzés forrását, s hogy az eset véletlen volt-e vagy szabotázs. Ez a rész ugyanúgy működik, mint a pénztárgépek „spiclidoboza", tehát adott ideig mindent tárol, és a felhasználó által nem módosítható, csak kiolvasható.

A kártya extra szolgáltatása a merevlemez valódi, fizikai írásvédelme. Általa a problémás programok tesztje vagy a csak olvasható adatbázisok védelme olyan egyszerű, mint amikor a 3,5"-es floppy fizikai írásvédelmét egy műanyag pecek eltolásával ki- vagy bekapcsoljuk. Minden szoftveres írásvédelmi rendszert valamilyen módon ki lehet játszani, csak türelem kérdése. (Ugyanezt teszik a vírusgyártó „kisiparosok" is.) A merevlemez fizikai írásvédelme viszont csak az írásvezeték megszakításával oldható meg. A TG kártya — ha már úgy is ott van — kihasználja ezt a lehetőséget.

A kártya vagyonvédelmi funkciója a Dataplan Rt. által kifejlesztett HISEC riasztórendszerhez kapcsolódik egy kiegészítő panelen keresztül. Ez a modul a kártyára opciónálisan kapcsolódik, csak azoknak kell megvenniük, akik külön igénylik.

A kártya azt is automatikusan érzékeli, hogy mikor kapcsolták ki a gépet. Ezt követően az első visszakapcsolásnál újra teljes vírusellenőrzést hajt végre. Mi több, a TG kártya azt is észleli, ha kiveszik a gépből. Ismételt visszahelyezés esetén „szól", és dokumentálja a műveletet.

Amennyiben a TG kártyából elegendő példány kerül forgalomba, a naplóállományok elektronikus összesítésével akár országos statisztikát is készíthetünk majd a vírushelyzetről, természetesen a felhasználókkal együttműködve. A különösen fontos adatok védelmére (pl. a Honvédelmi Minisztériumban, a Belügyminisztériumban) kiegészítésként valós idejű online adattitkosító modul is csatlakoztatható a rendszerhez.

A kártyából a prototípusok elkészültek, folyik azok tesztelése és a felületszerelt végleges változat áramkörének tervezése. Folyamatban van a vírusvédelmi eljárásoknak, illetve magának a kártyának a szabványosítása is. Az áramköri lap legyártása valószínűleg az USA-ban történik, az ottani MIL (az átlagosnál szigorúbb, katonai) szabványok szerint.

Az első sorozatban gyártott példányok legkésőbb 1991. szeptember elején megjelennek a piacon. Az eladási ár a szérianagyságtól is függ, és az előzetes piaci becslések alapján Magyarországon 12 000-15 000 forint között várható. (Összehasonlításképpen: egy jelszavas bejelentkező kártya, RAM és vírusvédelmi funkciók nélkül 16 000-18 000 forintba kerül.)

Szegedi Imre

DataEase INTERNATIONAL

A **DataEase** adatbázis-kezelőt azoknak ajánljuk, akik értik az: angol, dán, finn, francia, holland, izlandi, magyar, német, norvég, olasz, orosz, portugál, spanyol, svéd nyelvek valamelyikét, ugyanis a **DataEase International** terméke ezeken a nyelveken is tud.

A **DataEase** egy egyedi vagy többfelhasználós (LAN) adatbázis-alkalmazást fejlesztő rendszer DOS környezetben, azoknak, akik a saját szakmájuk szakértői, akik színvonalas alkalmazásokat kívánnak egy-két nap alatt létrehozni, akik egyszerű nyilvántartásokat készítenek munkájuk segítéséhez, vagy akár azoknak, akik a számítástechnika professzionális alkalmazói.

Angliában 1990-ben a PC-s relációs adatbázis-kezelők közül a vásárlók több mint 30 százaléka a **DataEase-t** választotta, jóval többen, mint akármelyik másik terméket.

A **DataEase International, Inc.** termékeinek magyarországi disztribútora a:

VT-SOFT Videoton Software Kft.
1033 Budapest, Vörösvári út 103-105.
Telefon: 180-3744
Telefax: 180-3750

VT-SOFT®
VIDEOTON SOFTWARE KFT

INFORMÁCIÓKÉRÉS: 11 ▲

FAN computer

XT, 286, 386, 486 SZÁMÍTÓGÉPEK
IGEN KEDVEZŐ ÁRON
18 HÓNAP GARANCIÁVAL

AT 40 MB MONOKRÓM	**68.000,– Ft**
AT 40 MB VGA	**95.000,– Ft**

Részegységek, mouse-ok, digitalizálótáblák, scannerek nagy választékban.

3,5 inches, 1,44 MB WINMORE floppylemez 1.100,– Ft/doboz

FAN Electronics Ltd

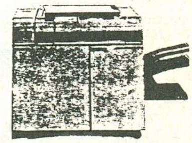
Tajvani—Magyar Vegyesvállalat
1118 Budapest, Késmárki u. 6.
(volt Friss István u.)
Telefon/fax: 185-0813

INFORMÁCIÓKÉRÉS: 14 ▲

Canon Canon

VÁLTSON SZÍNESRE
Canon
MÁSOLÓGÉPPEL!
CSÚCS AMIT TUD:

- Kicsinyítés
- Nagyítás
- Montírozás
- Tükörkép-készítés
- Képismétlés
- Poszter készítés
- 17 millió színáranyalat



Győződjön meg róla!

fénymásolás

TONER KFT
1095 Budapest, Mester utca 21.
Tel.: 113—1687, 134—3516

Canon Canon

INFORMÁCIÓKÉRÉS: 05 ▲

Különleges ajánlat az R-SOFT-TÓL!

Az első 20 vevőnk, aki nálunk vásárol

CLIPPER 5.0

adatbázis-kezelőt, egyéves ingyenes tagságot kap a „The European Nantucket Users Club"-ba!

Mit tartalmaz a csomag?

- CLIPPER 5.0 programcsomag (jogtiszta, regisztrálható)
- CLIPPER 5.01 upgrade

És mindezt 99 000 forintért!

HunClip — a magyarul beszélő Clipper. Minden Clipper-operátor, -parancs, -függvény a magyar ábécé szerint működik! A magyar nyelv teljes támogatása Clipperben! Egyetlen programsor megváltoztatása nélkül! Teljes kompatibilitás! Clipper Summer '87-hez és 5.0-hoz egyaránt kapható! Hihetetlen, de **ára csak 4980,– Ft + ÁFA**

Rendeljen még ma!

R-SOFT-SZENZOR Tanácsadó Kft.
H—1277 Budapest 23., Pf. 45.
Telefonszám: 201-6891. Telefaxszám: 201-8619

Megrendelem az R-SOFT-SZENZOR Tanácsadó Kft.-től

☐ a Clipper 5.0 programcsomagot 99 000,– Ft + ÁFA áron

☐ a HunClip Clipper-bővítést 4900,– Ft + ÁFA áron

(A megrendelni kívánt termék neve mellé tegyen egy x-et!)

Név: ....

Cím: ....

....

Ha **CLIPPER,**
akkor **R-SOFT—SZENZOR!**

INFORMÁCIÓKÉRÉS: 08 ▲

INFORMÁCIÓKÉRÉS: 17 ▼

Új rovat indul...

Olvasóink hiányolták, hogy egy-egy nagyobb cikkhez, sorozathoz vagy összeállításhoz miért nem közlünk — a Könyvespolc rovat mintájára — bibiliográfiát a folyóiratokban megjelenő szakcikkekről is. Az InfoNet Kft. most vállalkozott arra, hogy Sandokan elnevezésű számítástechnikai szakirodalmi adatbázisából rendszeresen ellátja ilyen válogatással az Alaplap szerkesztőségét, és annak egy részét most elindított Témabővítő rovatunkban közölhetjük. Első ilyen összeállításunk teljes egészében a hónap témájához kapcsolódik.

Az információk forrásának feldolgozásáról érdemes tudni, hogy a Számalk számítóközpontjának utódaként 1990. január elsején megalakult InfoNet cég munkatársai a Sandokan számítástechnikai adatbázis anyagát mintegy száz, nagyobbrészt nyugati folyóiratból válogatják, szerkesztik, lektorálják és tárgyszavazzák. A fontosabb cikkekről magyar nyelvű tömörített összefoglalás is készül. Az adatbázis havonta bővül, jelenleg több mint 50 ezer rekordot tartalmaz, és 8 csoportban mintegy 70 számítástechnikai szakterületet ölel fel.

Terjedelmi okokból az általunk közölt bibliográfia természetesen egy-egy témakörben is mindig csak töredéke lehet a Sandokan adatbázis teljes tartalmának, amely az InfoNetnél akár online módban, akár floppyn vagy nyomtatott formában hozzáférhető és előfizethető. (E szolgáltatásokkal kapcsolatban bővebb felvilágosítás: InfoNet Kft. 1111 Bp. XI., Vahot u. 6. Telefon: 166-9065.)

Vírusvédelem, adatbiztonság

Angol nyelvű cikkek

The Novell virus
(Novell hálózatok vírusai)
Computers & Security (NL) 1990/7

Random bits and bytes (A jelszavas hozzáférés problémái, alkalmazási trendje, számítógépbiztonság és vírusvédelem)
Computers & Security (NL) 1990/2 és 3

Friday the 13th, facts and fancies (A Péntek 13 vírus okozta hatások elemzése)
Computers & Security (NL) 1990/2

Computer virus prevention and containment on mainframes (Vírusfertőzés megelőzése és kezelése nagyszámítógépeknél)
Computers & Security (NL) 1990/2

The IPM model of computer virus management (A számítógépvírusok elleni küzdelem modellezése)
Computers & Security (NL) 1990/5

Microcomputer security systems attempt to lock up desktops (Az irodai desktop-rendszerek biztonságának jelentősége, biztonsági rendszerek)
Computers & Security (NL) 1990/2

Computer viruses (Számítógépvírusok és a megelőzés módszerei az IBM PC-knél — gyakorlati tanácsok)
Information Age (GB) 1990/2

Protection from viruses (Különböző típusú „perverz" szoftverek és az ellenük való védekezés módjai)
Information Age (GB) 1990/2

Handbook on how to handle viruses (Kézikönyv a vírusok kezeléséről)
Computerworld Australia (AU) 1990/32

Computer viruses: can they be prevented? (A számítógépek és programok vírusfertőzése elleni védekezés és a megelőzés módja)
Computers & Security (NL) 1990/2

Abstract of recent articles and literature (A számítógép-biztonság, az adat- és szoftvervédelem kérdéseivel foglalkozó cikkek kivonatai és bibliográfiájuk)
Computers & Security (NL) 1990/2 és 3

Security software (22 adatvédelmi, adatbiztonsági szoftver összehasonlító értékelése)
Which Computer? (GB) 1990/7

OSI and X.400 security (Biztonsági kérdések az OSI és az X.400 rendszerekkel kapcsolatban)
Telecommunications (US) 1990/5

Tools of the PC trade (Adatállomány-helyreállító és adatvédő szoftverekről)
PC user (GB) 1990/132

Safety first (Adatok biztonságát növelő, DOS-alapú programcsomagok)
PC user (GB) 1990/141

A secure future for diskless workstations (Lemez nélküli munkaállomások mint az adatvédelem megbízható eszközei)
Datamation (GB) 1990/22

Backing up (Két adathelyreállító DOS-segédprogram ismertetése)
What Micro? (GB) 1990/9

Keep your data secure (Adatvédelmi, adatbiztonsági szoftverek és megoldások)
Datamation (GB) 1990/23

Security architectures for textual databases (Szöveges adatbázisok biztonsági architektúrái)
Computers & Security (NL) 1990/7

The SeaView security model (Egy sokfelhasználós biztonsági rendszer modellje)
IEEE Transaction on Software Engineering (US) 1990/6

ABYSS: An Architecture for Software Protection (Egy szoftvervédelmi architektúra)
IEEE Transaction on Software Engineering (US) 1990/6

Using AI to improve security (Mesterséges intelligenciával az adatbiztonság növeléséért)
Datamation (GB) 1990/3

Német nyelvű cikkek

Datenverkehr ohne Risiko: Virenschutz für Festplatten (Merevlemezek vírusvédelmére szolgáló hardverkártyáról)
MC — Die Mikrocomputer-Zeitschrift (DE) 1991/2

Hilfsprogramme für den PC: Hilfe aus dem Werkzeugkasten (Személyi számítógépeken használható segédprogramok bemutatása)
Chip (DE) 1991/3

Vergleichstest: Fünf Backup-Programme (Mágneslemezes háttértárak tartalmáról biztonsági másolatot készítő backup szoftverek összehasonlító értékelése)
PC Welt (DE) 1991/3

Virulente Zeitbomben (Időzített vírusbombák)
Computer Persönlich (DE) 1990/8

Erste Hilfe für den PC („Elsősegély" programok személyi számítógépekhez)
PC Welt (DE) 1990/4

Sicherheit zuerst! („Datasafe 286" adatvédelmi és biztonsági koncepció személyi számítógépekhez)
PC Welt (DE) 1990/5

Mitstreiter (A legjobb Atari-ST segédprogramok)
Chip (DE) 1990/6

Viren-Impstoff per Stecker (Elektronikus vírusdetektor Atari-ST számítógépekhez)
Chip (DE) 1990/6

Sicherer Datenschutz der Soft- oder Hardware (Személyi adatok hardver- és szoftvervédelme személyi számítógépeken)
Computer Persönlich (DE) 1990/2

Algorithmen, Mechanismen und Dienste (Információközlő hálózatok biztonságát növelő algoritmusok, mechanizmusok és szolgáltatások)
IT — Informationstechnik (DE) 1990/1

Sichere Kommunikationsnetze und deren Protokolle (Biztonsági rendszer beépítése az adatátviteli hálózatok szabványaiba és az alkalmazásokba)
IT — Informationstechnik (DE) 1990/1

Achtung! Infektion! (Vigyázat! Vírusfertőzés! — gyakorlati tanácsok)
PC Praxis (DE) 1990/4

Utilities machen's einfacher (Húsz segédprogram bemutatása)
PC Welt (DE) 1990/12

Magyar nyelvű cikkek

Hogyan harcolunk az IBM-kompatibilis PC-kben a vírusok ellen? 1. rész
Chip (HU) 1990/3

Vegyük komolyan az adatbiztonságot
Computerworld-Számítástechnika (HU) 1990/20

A Turbo Anti Virus Toolkit tesztje
Computerworld-Számítástechnika (HU) 1990/35

Antivírus-programrendszerek
Computerworld-Számítástechnika (HU) 1990/27

Aktív védelem
Computerworld-Számítástechnika (HU) 1990/35

Összeállítva a Sandokan adatbázis alapján.
InfoNet Kft.
1111 Budapest
XI., Vahot u. 6.
Telefon: 166-9065

infoNET

Thunderbyte PC Immunizer
„A vírusok nem kártyakompatibilisak”

A vírusvédelemre használható ismertebb kártyák közé tartozik a Thunderbyte PC Immunizer. (A Systems and Special Services cég terméke.) Két korábbi változatát vettük meg és teszteltük le (v1.00 és v1.10), a legújabb 2.0 verziót pedig a cég európai disztribútorának standján, a budapesti Ifabo kiállításon volt módunk „megizzasztani”. A tesztnek volt egy érdekes mozzanata. A kártya programozója kijelentette, hogy azok a programvírusok, amelyekkel mi „átvertük” a kártyát, nem is vírusok. Így a Polimer, a Kukac és a Turbo Kukac azonosítóit a TBSCAN víruskereső újabb kiadásból ki is vették. Amivel nem tudunk megbirkózni, az nincs is?!

A Magyarországon kapható első Thunderbyte példány alaposan megnehezítette munkánkat, ékes holland nyelvű dokumentációjával, amelyet számítástechnikai szak-angoltudásunk birtokában kellett kínnal megfejtenünk. Az ismertető azonban ettől elvonatkoztatva sincs elég alaposan megírva.

A kártya vírusvédelmi rendszere a beépített BIOS-ban lévő szoftverrel és a lemezről installálható programmal működik. A TBINSTAL programot a kártyának a gépbe való behelyezése előtt kell elindítani. A program megállapítja a kártya BIOS-ának installálási címét és a kártya I/O címét. A kártyán lévő 1-es kapcsolósor (switch) egyes elemeit ennek megfelelően kell beállítani. A második kapcsolósor a kártya hardverlehetőségeinek aktivizálására szolgál. Csak a kapcsolók megfelelő beállítása után kell a Thunderbyte kártyát a gépbe helyezni.

E kártyával történő „együttélés” számítástechnikai szakismereteket és a víruslélektanban való jártasságot igényel. Fejlesztői szerint a kártya általános célú, tehát nem egyes vírusok ellen készült, hanem vírusfunkciók ellen nyújt védelmet, nem öl vírust, csak vírusfunkciókat detektál. A kártya BIOS-a a PC BIOS-a után fut le (de még a DOS előtt). A kártya védelmi szoftverét (TBCONFIG.COM) a CONFIG.SYS vagy az AUTOEXEC.BAT fájlból kell installálni a DOS betöltése során. A merevlemezre történő írás engedélyezése vagy tiltása a vezérlőkábel megszakításával lehetséges — a kártyát elektromosan a merevlemez és a kontroller közé iktatva —, és csak a kártya által engedélyezett műveleteknél hajtható végre.

A Thunderbyte kártya és a hozzá adott szoftver angol, német és holland nyelven képes kommunikálni. Az egyes üzenetek a kártyán lévő BIOS-ban vannak, így annak „locsogó” volta miatt vajmi kevés hely maradt az érdemi programokra. A kártya üzenetei nem mindig egyértelmű formában, de minden esetben felhasználói döntéshez kötötték a rendszer további működését. A felhasználónak magának kell(!) döntenie, de néhány esetben döntésétől függetlenül a vírus jön, lát és győz!

A kártya koncepciója: nem enged más programot rezidenssé válni, csak azt, amelyiket a meghatározott konfigurációs állományban (SET TB=xxxxxxxx.yyy) megadtunk neki. Nem enged továbbá a végrehajtható (EXE, COM, SYS) programokba írni, a kártya BIOS-ába fixen beégetett kiterjesztéseket figyeli. Akkor viszont mi a teendő, más programunk is lehet végrehajtható (mondjuk a GEM rendszer .APP állományai), és azokat is írásvédeni akarjuk.

Lehetőségünk van a CMOS setup információváltozásainak figyelésére, valamint többletszolgáltatásként a számítógépben ez jelszóval levédhető. A jelszót a merevlemez 0. track 6. szektorának elejére írja be, a 2. kapcsolósor állásával kódolva. Néhány boot-vírus is ír azonban erre a szektorra, s ha átmegy, akkor kesereghetünk kivasalt merevlemezünk felett. A szovjet eredetű Mirror ezt a szívességet készséggel megteszi nekünk, 5-10 másodperc alatt.

A beszerzett két kártyát az alábbi konfigurációjú gépen teszteltük, valódi „katalógusvírusokkal”: 80386-os AT, Phoenix BIOS, TVGA VGA monitorkártya, két merevlemez (egy 80 és egy 40 MB-os), 2 MB RAM, MS-DOS 4.01 BIGDOS partícióval. A kártya szoftverét (TBCONFIG) a CONFIG.SYS állományba helyeztük el. A rezidens program egyformán indítható device driverként a CONFIG.SYS-ből és programként az AUTOEXEC.BAT-ból, de

A Thunderbyte kártya és a vírusok

Teljesen véd az alábbi vírusok ellen:
Stoned, Ping-Pong '86, Töltögető, (c)Brain, Yankee Doodle, Vienna/648, Victor, Jerusalem, 1260, 4096, Mixer, Barcelona, Perfume, Syslock, Oropax, Time/Monxla, Joker, Devil's Dance, Invader/Plastiq (.EXE version).

Szerencsés döntés esetén véd az alábbiak ellen:
Form (boot), Eddie, V2000, April 1.

Kártyával ugyanúgy vagy még inkább fertőző vírusok:
Polimer, Invader/Plastiq (.COM verzió).

Kártyával is rendszerösszeomlást okozó vírusok:
Ogre/Disk killer, Print Screen, Invader/Plastique (boot rutinja), Vacsina, Alabama, W-513, Flip, Vcomm.

meghajtóprogramként jóval nagyobb a paraméterezési lehetőség, így mi ezt az utat jártuk végig. Mindezeken felül a FASTX extended memóriát kihasználó merevlemez-cache programot használtuk a diszkműveletek gyorsítására. A fertőzött programokat DOS-ból és a Norton Commander v3.00 alól indítottuk el.

A probléma már a Norton Commandernél kezdődött. Ha új alkönyvtárat akartunk létrehozani a floppyn, akkor a kártya rögtön szólt: bootszektor-változás történik (lehet hogy a cache program miatt?). Nem tudni, mi volt a kissé idegesítő és éppen nem helyénvaló üzenet. Ugyanez a probléma merült fel, ha a Norton Commander alól állományokat másoltunk ki a merevlemezről floppyra. Arról már nem is beszélve, hogy a hajlékonylemezek inicializálásához külön trükköket kellett alkalmazni, amolyan „jobb kezemmel megfogom a bal fülem" típusú bonyolult utakat. Közben a kártya folyton sipákolt.

Ígéretes új operációs rendszer a DOS 5.xx. Sajnos a Thunderbyte kártya általunk megvizsgált két eltérő verziószámú példánya közül egyik sem tud ezzel a DOS-sal működni. Nem tölthető be a DOS 5.00 operációs rendszer floppylemezről. Ha viszont az installált operációs rendszert merevlemezről töltjük be, akkor a kártya már a betöltés folyamatában vírust jelez, ami természetesen csak vaklárma. Ha nagy nehezen mégis betöltjük az operációs rendszert, akkor a kártya szoftvere nem érzékeli a kártyát.

Teszteléskor a boot-vírusok közül elsőnek a Stoned/Marijuana vírussal fertőzött lemezt tettünk az A: floppy egységbe. A kártya nagyon jól kivédte ezt a fertőzési kísérletet. Ránézett a floppyra, és ezt követően a C: merevlemezről betöltötte az operációs rendszert. Hasonlóképpen elhárította a (c)Brain floppyfertőző boot-vírust is. Utánna az Alt billentyű lenyomásával a kártyát kikapcsoltuk, és megfertőztük a gépet a Stoned/Marijuana vírussal. Ekkor a kártyát visszakapcsolva és a rendszert újraindítva a DOS betöltési folyamata a CONFIG.SYS fájlnál leállt (valószínűleg a TBCONFIG.COM program memória-allokációs gondja miatt), és a gép lefagyott. A gépet csak a kártya kiiktatása és a vírus kiirtása után lehetett ismételten használni. Komoly hiba! (Ki dolgozik a felhasználók közül mindig „tüzelésre kész" csavarhúzóval?)

A Print Screen boot-vírussal tesztünk során megfertőztünk egy nem rendszerlemezt. (Ez a boot-vírus Magyarországon most van terjedőben.) Ha erről a lemezről akartunk operációs rendszert betölteni — mint amikor a figyelmetlen felhasználó a floppyt benne felejti a meghajtóban —, akkor a BIOS (Phoenix) az F1 billentyűvel a rendszer újraindítását kérte, amitől le is fagyott a gép. Ugyanezt tapasztaltuk az Ogre/Disk Killer esetében.

A fájlvírusok egy részével szembeni tehetetlenség oka a program írójával, Frans Veldmannal folytatott beszélgetésből derült ki. A kártyát ugyanis nem éles vírusokkal, hanem vírusszimulátorokkal tesztelték. Így azután nem csoda, ha „a kártya nem teljesen víruskompatibilis", nem mindegyik ellen hatásos. Máskor olyan víruseffektusok lépnek fel, ami a vírusok forráskódjának ismeretében is alapos meglepetést okoz. A magyar Polimer vírus például vígan elsétált mellette. Az Eddie/Dark Avenger vírust kifejezetten doppingolja a kártya: normális esetben Eddie csak egyszer épül be egy állományba, a kártya megléte esetén pedig akárhányszor. A Victor/Iván v1.0 verzióját korrekten felfedezi, és a fertőzést megakadályozza. Hasonlóképpen viselkedik a Jerusalem/PLO vírussal és a Vienna/648-cal is.

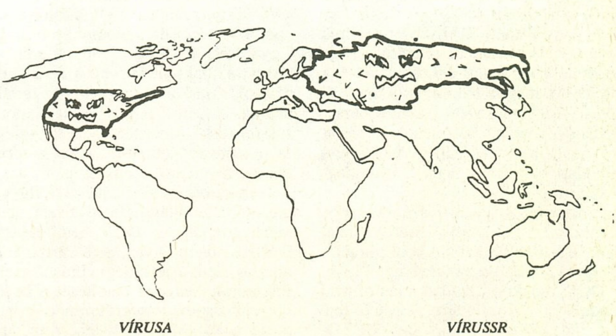
Érdekes a Vacsinával való bensőséges kapcsolata. Ha Norton Commander alól futtatjuk a vírusos programot, akkor jelzi az INT 21 cseréjét. Ennek nem engedélyezése esetén a vírus sem fertőz. Ha rossz gombot nyomva mégis engedélyezzük, akkor a kártya folyamatosan vonyít, és a rendszer leáll, csak a főkapcsolóval lehelhető életre. Ha nem a Norton Commander alól indítjuk a programot, más a helyzet. A kártya szintén jelzi az interruptcserét, de az engedélyező válaszra a DOS „Devide overflow" rendszerüzenettel kiakad. De legalább befogja a hangszóróját!

Az új generációs vírusok közé tartozik a 4096, amely sajnos egyszerre több verzióban felbukkant Dél-Magyarországon. A kártya ennek a fertőzését kivédi. Nem szól semmit, s bár a vírus rezidenssé válik, nem fertőz tovább. Ez lényeges pozitívum.

Az éles vírusokkal folytatott kínvallatás után a kártyát kiadtuk fejlesztőinknek, hogy a szokásos körülmények között teszteljék működését. Egyikük a kártyával egy hétig kínlódott, míg a másik hamarabb feladta. Véleményük az volt, hogy a kártya részötleteiben jó, de nem felhasználóbarát, minden vírusgyanús esetben a felhasználónak kell eldöntenie, hogy engedélyezi-e az adott műveletet vagy sem. (A felhasználó sokszor még akkor sem tud dönteni, ha jól beszél angolul vagy németül, és az üzenetek pontosan vannak megfogalmazva.) A kártya jelenlegi tudásával a gyakorlatban nehézkesen használható, nem elég megbízható, és hamis biztonságérzetet kelt. Egyes vírusok működését csak részben vagy egyáltalán nem akadályozza meg, néhányét pedig még serkenti is.

Tesztünk eredményét elküldtük a fejlesztő cégnek is. Első reakciójuk dacos válasz volt a vírusok kártyakompatibilitásáról... De utána mégis felhasználták annak információit a fejlesztéshez, és megjelent az EXIT_FIX.COM rezidens program (a Thunderbyte 1.00 és 1.10 verzióihoz), amellyel a rendszerlemerevedéseket igyekeznek kiküszöbölni. Mindemellett a fejlesztő cég rendelkezésünkre bocsátotta a Thunderbyte v2.00 message compilert, és kérte a véleményünket. Ugyanis itt a rendszerüzeneteket már a szoftver tartalmazza, a disztribútorok fordítják be oda az adott ország nyelvén. De a kártyán sajnos továbbra is átmennek az eddigi „kártyajáró" vírusok...

Szegedi Imre



VÍRUSA *VÍRUSSR*

Kártyaparti melankóliában szenvedőknek

1990 novemberétől több forgalmazónál megjelent a Virus Guardian tajvani vírusvédelmi kártya. A kártya izraeli fejlesztés, amelyben ott nem láttak fantáziát, ezért az addigi eredményeket eladták Tajvannak.

A Virus Guardian kártya 8 és 16 bites slottal rendelkező gépekbe is behelyezhető. Szimpatikus benne az, hogy a kártyán akkumulátor van, ami feltételezi, hogy bizonyos alapinformációkat a gép kikapcsolása után is megőriz. Első pillantásra azt hihetnénk, hogy a kártya egy hosszú, speciális IC-t tartalmaz. Közelebbről megvizsgálva azonban kiderült, hogy a tajvani gyártó egy fémlappal leragasztotta a kártya BIOS-át és a kártyán lévő CMOS-t, mintegy véglegesítve a kártya szoftverét is. Ezzel nem tudták ugyan megakadályozni a ROM tartalmának kimásolását és visszafejtését, de a felhasználók zsebéből kilopják a pénzt, mert aki át akar térni egy fejlettebb változatra, újra meg kell vásárolnia az egész kártyát, nem csupán az új BIOS-t és a szoftvert, amire valójában szüksége lenne.

A tesztelés során behelyeztük a kártyát egy IBM-kompatibilis számítógépbe. (AT 286, 40 MB-os merevlemez, EGA kártya, 5,25" és 3,5" floppyegységek.) A rendszerindítás után mindjárt be is jelentkezett a kártya.

Első lépésben a kártya jelszavát kellett megadni. (Mi a kártya sorozatszámát használtuk erre.) A jelszót minden újabb rendszerindításnál be kell gépelni, mert a kártya követeli. Ha nem adunk meg jelszót, a kártya akkor is bejelentkezik, és ebben az esetben (ha nincs jelszó) az <Enter> billentyűvel mindig tovább lehet menni a kártya kérdésein.

Nem mondhatjuk el, hogy a Virus Guardian láthatatlanul működik. Függetlenül attól, hogy szükség van-e a kijelzésre, megpróbál látványosan működni. Melankóliában szenvedőknek hatásos orvosság lehet, másokat viszont örökös üzeneteivel, sípolásával rövid idő alatt bejuttathat valamilyen ideg-osztályra.

A kártyát olyan „új" számítógépbe helyeztük be, amelyet szoftverekkel még nem töltöttünk fel. A konfigurációt elvégezve a kártya megjegyezte, hogy a DOS mekkora szabad memóriával rendelkezik, és ezt eltárolta CMOS-ába. Ezt követően amikor az AUTOEXEC.BAT fájlban elhelyeztük a DOS rezidens APPEND parancsát, a kártya minden rendszerindításkor visított. Ha kivettük az APPEND parancsot, a visítás megszűnt. Végképp nem értjük, miért tekinti őt vírusnak. Hasonlóan viselkedik a GRAFTABL és a PRINT rezidens DOS-utasításra is.

A gépnek szoftverrel történő feltöltése során létrehoztunk néhány alkönyvtárat. Ahogy az elő szokott fordulni, nem minden állományt oda másoltunk be, ahová szerettük volna. Ezeket az állományokat a Norton Commanderrel (v3.00) átmásoltuk a kívánt alkönyvtárba, és a másolatot töröltük, tehát nem a Move parancsot használtuk. A VG kártya azonban nem hagyja olyan egyszerűen törölni az általa figyelt COM, EXE, SYS állományokat, hanem minden törlésre kétszer rákérdez a szokásos visítozás kíséretében. (60 állománytörlés = 120 visítás + 120 Y billentyű.)

A programokkal feltöltött gép használatakor kiderült, hogy a kártya egyes szoftvereknél érthetetlen módon lemerevíti a számítógépet. A Turbo Antivirus toolkitjéből a TNVIRUS.EXE programot szerettük volna működtetni, de a rendszer lefagyott. Ugyanennek a programcsomagnak a rezidens vírusvédelmi része (TSAFE) visítások közepette ugyan, de működésbe lépett. Ez egy kicsit meglepett minket. Ismét elindítottuk a TNVIRUS programot, és csoda történt, a program minden további nélkül elindult. Ezt követően minden olyan program indítása előtt, amely előzőleg lefagyott, a TSAFE programot indítottuk el. Kitűnően futottak...

Amikor a boot-vírusokat vizsgáltuk, a VG kártya az általunk elindított összes boot-vírust kivédte, néhány apró megjegyzéssel. A Töltögető eljutott a pusztító rutin által kiírt szövegig, de a vírus nem okozott kárt a merevlemez tartalmában. A Den-Zuk is eljutott a „non-system disk" rendszerüzenet kiírásáig, és a rendszer lefagyott, tehát a vírus átjutott a védelmi rendszeren.

Ha a kártya boot-vírust észlel, figyelmezteti rá a felhasználót, és tiszta rendszerlemezt kér a DOS betöltéséhez. Eddig rendben is van, de a kártya nem minden esetben ad hidegindítást, hanem folytatja a rendszer betöltését a behelyezett második lemezről minden ellenőrzés nélkül. Márpedig, ahol vírus van, ott lehet több is. A kártya pedig nem mindig tudja eldönteni, hogy a behelyezett operációs rendszerlemez fertőzött-e vagy sem. A kártya ebben az esetben is szól, hogy valaki a megadott számú fizikai merevlemezre akar írni — de akkor már késő, a vírus odaért...

A fájlvírusok közül a VG kártyát először is az egyik legveszélyesebbel, az Invaderrel teszteltük le, amelyik ellen a Thunderbyte kártya sem véd kielégítően. A VG kártya észleli, hogy valaki rezidenssé akar válni, és figyelmeztet erre. De közben az Invader már bele is írt a boot-szektorba. Ezt a fertőzést csak akkor vettük észre, amikor egy lefagyáskor ismét rendszert kellett indítani a merevlemezről. Természetesen a kártya érzékelte a boot-vírust, és nem engedte a rendszer újraindítását. (Nem szól, de véd a fertőzés után. És jöhet a csavarhúzó, a szerelés, majd a vírusirtás...) Ez a fejlesztők öngólja.

A VG kártya fájlvírusok elleni védelmi elvét a következő módon fogalmazhatnánk meg: „engedem, hogy a vírus rezidenssé, aktívvá váljon, de nem engedem, hogy tovább fertőzzön". A kártyát a szokásos fájlvírusokkal teszteltük. Meglepő volt, hogy a kártya néhány „közönséges" fájlvírusnál is érdekesen viselkedett. A fejlesztők arra sem gondoltak, hogy a grafikus szoftverek mellett is lehetnek vírusok, és a kártya üzenetei teljesen tönkreteszik a grafikus képernyőt. Az MS-DOS 4.01 DOSSHELL képernyőjét is sikerült absztrakt mintákkal kidekorálnia.

Az 1701/Cascade vírussal fertőzött programot Norton Commander alól elindítva a rendszer egyszerűen lefagyott. A 1704-es futtatása után pedig a kurzor folyamatosan szaladgált a képernyőn, mintha végtelen ciklusba esett volna. Egy klasszikus baci kiakasztotta a rendszert, s ezt nem vártuk tőle! A 648/Vienna elindítása után pedig a Write protect disk üzenetet kapjuk meg.

Teljesen véd a VG kártya a következő vírusokkal szemben: Stoned, Ogre/Disk killer, Print Screen, Ping-Pong '86, Form boot, (C)Brain.

Részben véd a Töltögető ellen.

Nem véd, illetve rendszerösszeomlást okoz az alábbi vírusok esetén: Invader/Plastiq boot rutin, Den Zuk, 1701/1704 Cascade.

Szegedi Imre

Computer MÁNIA

1991/3-4

IBM, AMIGA, C64, SPECTRUM



SEARCH FOR THE KING

BLUE MAX

FANTASY
Kalandok és sárkányok /AD & D/

DEMO

KILLING CLOUD

159 forint

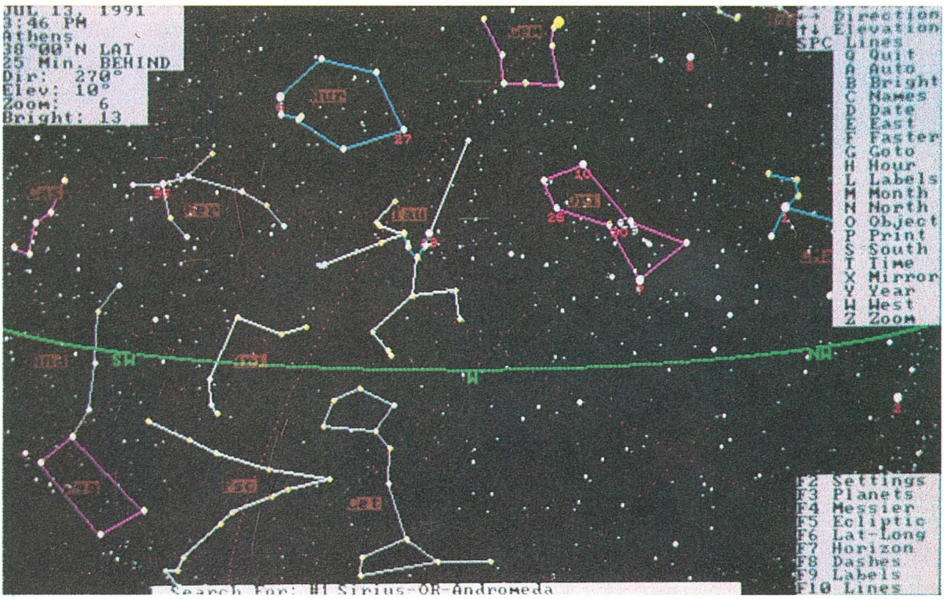
Skyglobe

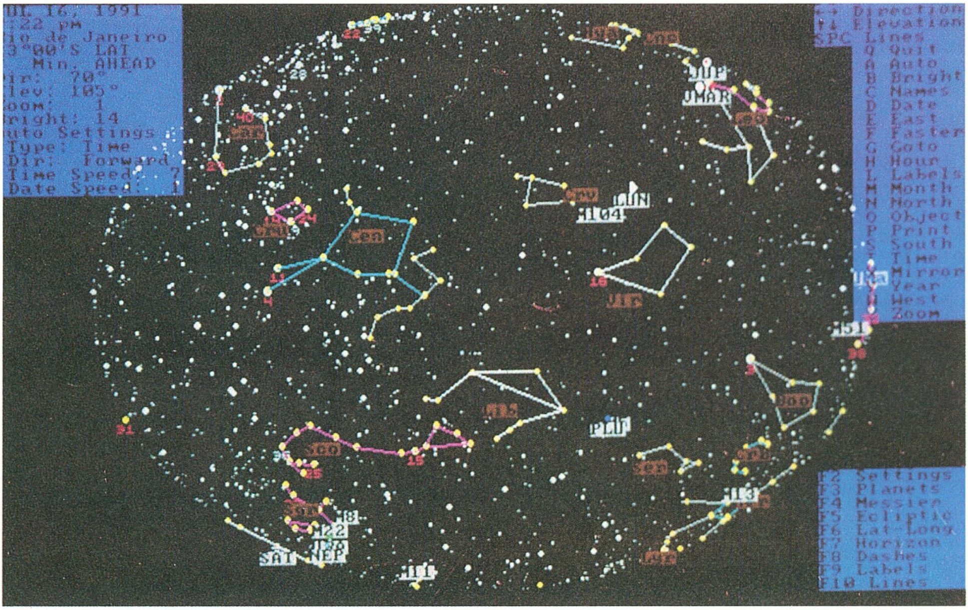
PC-planetárium

Nincs mindig lehetőségünk arra, hogy az éjszakai égboltot a szabadban tanulmányozzuk. Akik pedig nagyvárosban élnek, szinte alig látják a csillagokat. Egy nagyszerű shareware szoftver, a Skyglobe némi kárpótlást adhat. Mindazoknak ajánljuk, akik szeretnek időnként „világegyetemesdit" játszani, vagy egyszerűen csak érdeklődnek a csillagképek, az égi jelenségek iránt. A Skyglobe egy miniplanetáriumot varázsol személyi számítógépünk képernyőjére, és az égitestek látszólagos mozgását is nagy testvéréhez hasonlóan reprodukálja — a látványt pedig magunk tudjuk vezérelni.

A SKYGLOBE.COM elindítása után először be kell állítani az általunk használt képernyő típusát. Működik ugyan CGA és Hercules monitorral és kártyával is, de igazán élvezetes képet csak az EGA és a VGA grafikus megjelenítés nyújt. Utána azt a földrajzi pontot kell kijelölnünk, ahonnan az égboltot szemlélni akarjuk. Sajnos Budapest nincs a 120 megadott hely között, de nem sok az eltérés (legalábbis ebben a vonatkozásban), ha a közel azonos délkörön fekvő Párizst választjuk. A városok zöme amerikai, nekünk igazán érdekes kirándulást mégis inkább a déli féltekén lévő nézőpontok kínálnak, beleértve a déli sarkot is, mert az ottani égbolt egy részét innen Európából soha nem láthatjuk. De nem csupán a helyet, hanem az időpontot is megválaszthatjuk, messzire kalandozva a múltba és a jövőbe.

Úgy vonulnak el képernyőnkön a csillagképek, ahogy azokat a Föld forgásának megfelelően egyébként is láthatnánk. A mozgást gyorsíthatjuk és lassíthatjuk, de meg is állíthatjuk vagy visszafelé irányíthatjuk. Szabályozható a függőleges látószög, tehát hogy mennyire nézünk felfelé, akár akrobatikus 180 fokban is, bár 90 fok felett természetesen jobban megfelel látásmódunknak a vízszintes irányt módo-





sítani, akár finom lépésekben, akár egyetlen billentyűváltással, égtájak szerint.

A program EGA képernyőn 15.000, VGA-n 25.000 csillagot tud megjeleníteni, mégpedig azok fényességi fokozata alapján, tehát alacsonyabb érték beállítása esetén csak a valóságban is legfényesebbeket mutatja be (1 = 200 csillag). Módunkban áll továbbá a látómezőt 16 fokozat szerint beállítani, a totál képtől kezdve az égbolt egy-egy részének többszörös kinagyításáig.

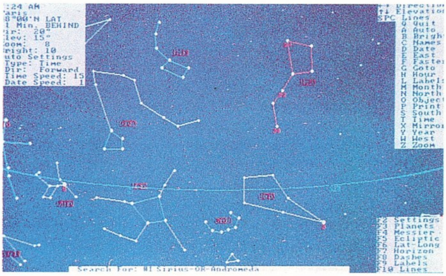
A bolygók mozgása szintén megjeleníthető, de miután ezek a csillagokkal ellentétben állandóan változtatják helyüket, a shareware program csak 1994-ig kezeli őket, mert minden év bolygómozgását külön .DAT állományban kell mellékelni, s egyetlen lemezre nem fért rá több.

ADATLAP
Név: Skyglobe
Szerző: Mark A. Haney, Klassm Software, USA, 1990
Leírás: égbolttérkép, planetárium.
Konfiguráció: DOS 2.0 felett, Hercules, CGA, EGA vagy VGA monitor.
SolarSoft lemezszám: 466.

Előfordul, hogy szeretnénk megkeresni az égbolton egy csillagot vagy csillagképet. Ezt eleve megkönnyíti a csillagok és csillagképek mellett opcionálisan feltüntetett nemzetközi névrövidítés, de a Skyglobe egyszerű keresőparancsa bármelyik képállásból lehetővé teszi a 300 legfényesebb csillag és a 89 csillagkép behívását. Még arra is mód van, hogy az „S" (star) és a „C" (constellation) mellé a Shift és az Alt billentyűkombinációt használva megkapjuk a megfigyelésére legkedvezőbb dátumot és időpontot.

Az égbolton való tájékozódás segédvonalait (horizont, nappálya stb.) és a feliratokat, neveket, menüket tetszés szerint egyetlen billentyűvel ki- vagy bekapcsolhatjuk. A képernyőn megjelenő csillagtérképeket pedig akár mátrixnyomtatóval, akár lézerprinteren kinyomtathatjuk.

Nagyon szimpatikusak a szoftver felhasználására és terjesztésére vonatkozó részek. A szerzők mindenkit arra biztatnak, hogy a programot adják tovább, jusson az el minél több emberhez.



Mindössze annyit kérnek, hogy a szoftver integritásának megőrzése érdekében minden hozzá tartozó fájlt együtt és változatás nélkül használjanak. Mód van természetesen a Skyglobe szabadszoftver regisztráltatására is. A 15 dolláros díj fejében olyan példányt kaphatunk, amelynek alapértelmezésébe beírják lakóhelyünket és annak földrajzi koordinátáit. Adnak hozzá továbbá egy számozott Skyglobe referenciakártyát, egy adatállományt 25.000 csillagról, és külön adatlemezt 16 év planetáriumi paramétereivel. Akinek még ez is kevés, az lemezenként 5 dollárért megkaphatja az 1900 és 2099 közötti időszak adatait is. (Egy 5,25"-es lemezre 18 év fér rá.)

Faklen Pál

Muzsikálj, PC!

Szoftver a SoundBlasterhez

A CMS SoundBlaster
a jelenleg hozzáférhető hangkártyák közül
az egyik legkiválóbb.
Népszerűségét nem utolsósorban annak köszönheti,
hogy teljesen kompatibilis az AdLib kártyával.

„Digital Delay"

Visszhangot, illetve késleltetést rendel egy mintához. Mivel az effektus erőssége paraméterezhető, a „telítettebb" hangzástól egészen a visszhangig terjedő széles skálából választhatunk.

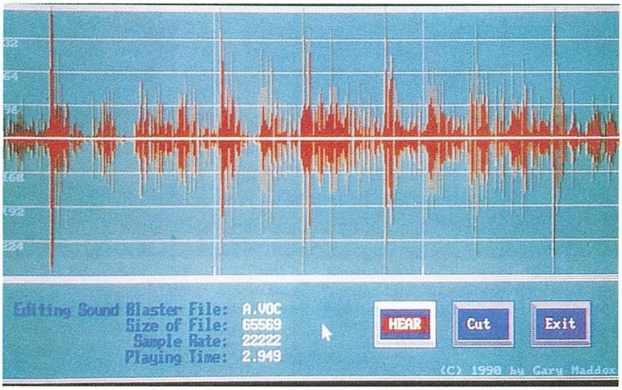
A hasonló kategóriába tartozó kártyákkal összehasonlítva a SoundBlaster különlegessége az, hogy alkalmas digitalizált hangok, dallamok, zenedarabok lejátszására. Ezeket az úgynevezett mintákat egy speciális bemeneten keresztül mikrofonról vagy sztereo berendezésről játszhatjuk be, ezután a szoftver feladata a felvétel és a visszajátszás. Sajnálatos módon a kártyához tartozó VOXKIT program lehetőségei eléggé korlátozottak. Hiába is próbálnánk egy-egy minta meghatározott részleteit kivenni vagy utólag feldolgozni, ezt sajnos nem teszi lehetővé a VOXKIT.

Lemezmellékletünkön mintapéldákat találhatnak arra, hogyan lehet saját szoftvert készíteni a SoundBlasterhez. Biztosak vagyunk azonban abban, hogy a SoundBlaster kártya tulajdonosai közül nem lesz mindenkinek elegendő ideje, kitartása, esetleg hozzáértése ahhoz, hogy önmaga oldja meg ezeket a feladatokat. Ezért néztünk körül a shareware-piacon, melyek lennének azok a programok, amelyeket bárki felhasználhat.

Az első ilyen jellegű program a „Blaster Master". Ennek rövid ismertetőjében is számos olyan lehetőséget ígér a szerző, amely mindeddig csak az igen drága „Digital Audio Workstations" tulajdonosai számára volt elérhető.

Az első kényelmetlenség a „Blaster Master" használatakor: csakis olyan mintákat képes feldolgozni, amelyeket „egyben" talál meg. (Ez angolul: RAW AND UNCOMPRESSED.) Tudjuk, hogy a SoundBlaster-minták adatformátuma (VOC) számos egyszerűsítési lehetőséget kínál. Ilyenek például az adattömörítés, a szünetek tömörítése, az ismétlési ciklusok beállítása, a kijelölések beállítása, ASCII formátumú szövegek beiktatása.

Látható tehát, hogy igencsak bonyolult szerkezetekkel is találkozhatunk.



A „Blaster Master" csakis azokkal a VOC állományokat tudja kezelni, amelyeket nem szerkesztettünk át utólag, tehát ahol nem használtuk ki a fenti lehetőségeket.

A „Blaster Master" shareware-verziója is kétlemezes, olyan programok gyűjteménye, amelyek alkalmasak meglévő VOC állományok manipulálására.

A „Blaster Master" főprogram tulajdonképpen nem más, mint egy menü, amely az alprogramokat aktivizálja.

Nézzük tehát a lényeget, melyek is ezek az alprogramok!

„Slow It Down!"

Ez a program kétszeresére növeli az adott VOC állomány lejátszási idejét, anélkül, hogy a hangmagasságot (pitch) módosítaná. A minta tehát lassabban fog szólni, de nem úgy, mint a lemezjátszónál vagy magnónál megszoktuk. Nem mélyül a hangszín, hanem ugyanolyan marad, mint az eredeti.

„Volume Control"

Erősíti vagy csökkenti a VOC állomány amplitúdóját. Ez azt jelenti, hogy a kérdéses minta hangosabban vagy halkabban fog megszólalni.

A gyakorlatban azt tapasztaljuk, hogy ha egy mintát sztereóról viszünk be, akkor a SoundBlaster bemenetének nagyfokú érzékenysége miatt halkan fog csak megszólalni. Ezzel a programmal utólagos hangosítás oldható meg.

„The Mixer"

Két mintából egy új mintaállományt kever. (Sajnos ez a funkció nem mindig működik!)

„Reverser"

Megfordítja a kérdéses VOC állomány lejátszását, azaz visszafelé játszhatjuk le az állományt.

„Hi-pass/Lo-pass Filter"

Szűrőt (filter) állít a mély (Hi-pass), illetve a magas (Lo-pass) frekvenciák-

ra. Paraméterként a „mélységet", azaz a szűrő erősségét adhatjuk meg.

„Hi-Res Sampler" (csak 80386-os gépeknél)

13 kHz-nél magasabb frekvenciával is engedélyezi a felvételt. Normál esetben a VOXKIT felső határa 13 kHz. 22 kHz-es értéket is realizálhatunk. Ez a program azonban csakis DOS-ból hívható!

„Down Sample"

Nagy mintavételezési frekvenciával felvett VOC állományok esetén használjuk. Jóval több memóriaterület szükséges az olyan állományok tárolására, amelyeket nagy mintavételezési frekvenciával vettünk fel, mint amelyeket alacsonyabbal. Egyes PC-knél gond lehet a nagyobb frekvenciával felvett állományok lejátszása is. A gyakorlat azt mutatja, hogy érdemes nagyobb frekvenciával felvennünk, majd pedig ezt leszállítani. Ilyenkor ugyanis a hangminőség sokkal jobb lesz, mint ha ugyanolyan, alacsonyabb frekvenciával történt volna a felvétel is.

„Sound Graph" (csak EGA/VGA kártyáknál)

Grafikusan jeleníti meg a VOC állományt. A PCM állományok különböző felbontásokkal jeleníthetők meg.

„Sound Edit" (csak EGA/VGA kártyákkal, egér is szükséges)

A grafikus megjelenítés mellett az állomány szabdalását is lehetővé teszi. Ez azt jelenti, hogy a zavaró kezdő- és végjelek gyorsan eltávolíthatók. Sajnos a program kezelése nem igazán kényelmes. Ennek ellenére azért alkalmas arra, hogy kiszűrjük vele az adott minta meghatározó részét. Ezt tulajdonképpen már a VOXKIT-től is elvártuk volna, hiszen állandóan szükségünk van ilyen műveletekre.

„Blaster Scope"

„Real-time" mintabevitelt mutat. Van EGA/VGA, illetve CGA/EGA/VGA verziója is, ezek azonban nem mindig működnek tökéletesen.

Az eddig ismertetett alprogramokon kívül azt is lehetővé teszi a „Blaster Master", hogy VOC állományokat közvetlenül (a CT-VOICE.DRV meghajtó nélkül) is lejátszhassunk, illetve felvehessünk. Felvételkor a mintahossz és a játszás ideje megjelenik.

A shareware-verzió egyik korlátja az, hogy legfeljebb 64 kilobájtosak lehetnek a minták, amelyeket felveszünk, illetve feldolgozunk. Ez azonban a teszteléshez mindenképpen elég.

Összefoglalva elmondhatjuk, hogy a „Blaster Master" minden korlátja ellenére a legjobb olyan shareware program, amely a SoundBlaster kártyák tulajdonosai számára a felvételt, visszajátszást és a minták feldolgozását lehetővé teszi. Azt azonban ne várjuk, hogy koncepcióját tekintve versenyezhet például egy olyan jól átgondolt és kényelmes programmal, mint amilyen az Amigára készült „Audiomaster"!

Verebély Pálné
A DOS Shareware cikke alapján (1991/3.)

Magyar shareware-katalógus

No.	Programnév	Megjegyzés	Darab	Tartalom
M001	Vírusvédelem	1	2	Vírusdetektor, ORCAD állományok konverziója
M002	SIMON	K	1	Matematikai problémák folytonos idejű szimulációja
M003	DIGGER-Editor		1	Pályatervező az ismert DIGGER játékprogramhoz
M004	TED	*	1	Zseniális C, PASCAL és Clipper programeditor!
M005	ALGEBRA	K	1	Numerikus algebra; mátrixok interaktív megoldása
M006	KATARC		1	Könyv-, folyóirat-katalogizáló és visszakereső
M007	CÍMJEGYZÉK		1	Név/cím/telefonjegyzék, körlevél/címkekészítés
M008	UNI-FORM		1	dBASE képernyőtervező, egyszerű kódgenerátor
M009	Turbo C Toolkit	K	1	Cache, menü, B-tree adatb.-kez., screen-generátor
M010	MSC Toolkit	K	1	Cache, menü, B-tree adatb.-kez., screen-generátor
M011	Sybilla Spring	K	1	Hálózatos(!) jelszavas határidőnapló
M012	JOKER		1	Minősítő és rangsoroló program (döntés-előkészítés)
M013	MANAGER	K	1	Irodaigazgatási rendszer (FoxBase-ben)
M014	MASA	K	2	Pénzügyi és számviteli rendszer (3 nyelvű!!!)
M015	AFA	K	2	Integrált ügyviteli rendszer (ÁFA + folyószámla)
M016	MAKELABELI	K	1	Szabad formátumú univerzális címkenyomtató
M017	Nagy Machinátor	K	2	Általános célú ügyviteli és menedzserprogram
M018	Magyar Ékezetek		1	Fonteditor nyomtatóra, EGA/VGA-ra, billentyűátdef.
M019	TRACESHOW		1	Rezidens DOS-handler nyomkövető EGA/VGA-hoz
M020	PC-JOG		2!	Hatályos jogszabályok 1945-től napjainkig!
M021	DIRI 5.01	K	1!	Vezetői-titkársági direkt információs rendszer
M022	BEX		1	BAT—EXE konverter, ablaktechnika, kiegészítések
M023	KEYBDRV		1	Teljes magyar ékezetesítő program (prin/mon/keyb)
M024	BLISS		1	Korlátozás nélküli főkönyvi rendszer C-ben!
M025	LEXICON	K	1	Szakfolyóiratok témafigyelése
M026	LETMARK	K	1	Postaforgalom iktatása
M027	Szociometria		1	Kisközösségi kiértékelő rendszer
M028	SURA	K	1!	Univerzális rajzolóprogram, C/PASCAL interfész
M029	OOplus		1	Adatbázis-kezelés TP 5.5-ben (objektumorientált!)
M030	CAB		1	Pénzügyi matematika Lotus 1-2-3 WK1 formátumban
M031	TPUDISASM		1	Turbo PASCAL 5.x TPU visszafejtő
M032	VÁM91		1	Vámáru-nyilakozati űrlapot kitöltő program
M033	PASSIV DEMO		1	Interaktív német nyelvtani tesztprogram
M034	HELPEDIT		1	Online help készítése CLIPPER-es programokhoz
M035	TDBU (TURBO DBU)		1	Interaktív DBF-kompatibilis adatbázis-kezelő
M036	FILECAT		1	Lemeztartalom-nyilvántartó és -visszakereső
M037	POSTAFORGALOM		1	Kisvállalat napi postájának iktatása, szignálása
M038	MANDELBROT		1	Esztétikus színes ábrák +
M039	SPT_GIB DEMO	K	1	Grafikus angol–magyar, magyar–angol szótár demója
M040	GREAT SPECULATOR	K	1	Interaktív brókeri és titkársági szoftver

Jelmagyarázat:
* — A FLOPPYLAND-ben azonnal regisztrálható!
! — 1,2 Mbájtos floppyval, lemezenként + 100 Ft felárral
K — Korlátozott képességű, demónak tekinthető

Várjuk minden olyan vállalkozó kedvű magyar programozó jelentkezését, aki arra érdemes szoftvertermékeit szeretné nyilvánosan is megmérettetni, ötleteit hajlandó másokkal is megosztani, illetve szeretné élni a shareware-koncepció adta ingyenes reklámlehetőséggel.

Tegl Windows Toolkit II

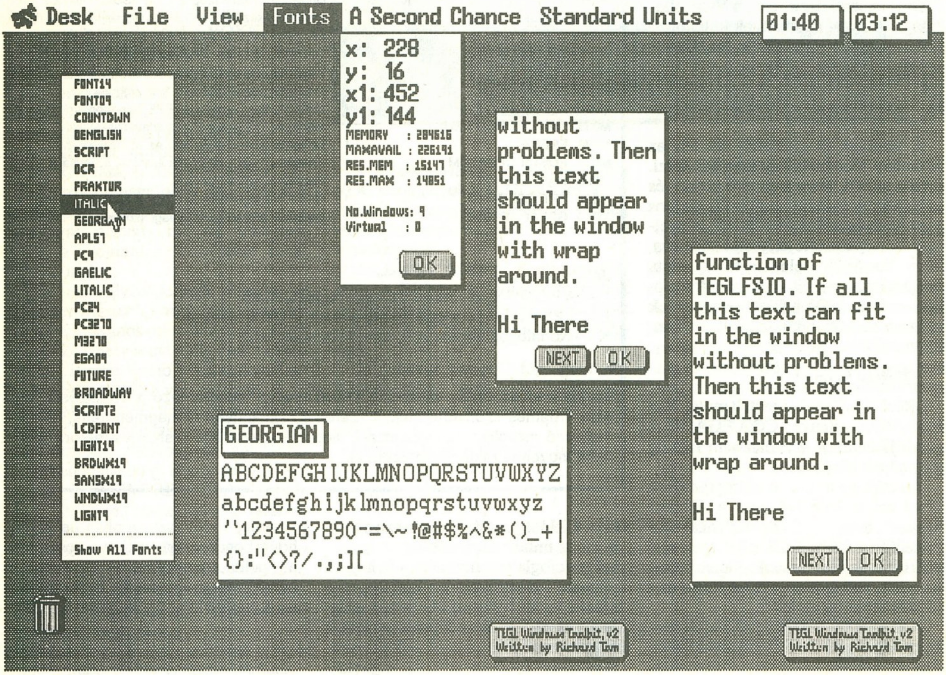
Az egyéniség diadala

A Tegl Windows Toolkit II megjelenésével nem jelent többé gondot a Turbo Pascal 5.0/5.5, Quick Pascal 1.0, Turbo C 2.0, Quick C 1.0 és az MSC 5.1 nyelveken programozók számára a grafikus környezetben működő programok fejlesztése. A Toolkit a grafikus ablakok, a teljes körű egértámogatás mellett ikoneditort is tartalmaz, ezenkívül különböző betűkészleteket használhatunk szöveges megjelenítéskor.

A Tegl támogatja az összes elterjedt grafikus kártyát (Hercules, CGA, EGA, VGA és Super VGA). Maximum 16 színnel dolgozhatunk (EGA-tól felfelé). Mindig az adott grafikus kártya maximális felbontását támogatja, egészen az SVGA 800X600-as maximális felbontásáig. Nagyon kényelmessé teszi a programozást, hiszen csakis magas szintű procedurákat kell használnunk. Így minimálisra csökken az a pluszmunka, amely ahhoz szükséges, hogy a Toolkit alkalmazását elsajátítsuk. Miután bizonyos gyakorlatot szereztünk, alacsony szintű területekhez is hozzá tudunk férni, így a Toolkit által nyújtott lehetőségek teljes skáláját teljes mélységben ki tudjuk használni. Tulajdonképpen csak azt kell megadnunk, hogy mi hol jelenjen meg ikonként, illetve menüként, milyen procedurákat (vagy függvényeket) akarunk meghívni akkor, ha az adott menüpontra kattint majd rá az egérrel a felhasználó. A rendszerhez tartozó ikoneditor segítségével igen egyszerűvé válik az ikonok definiálása, ezeket később programjainkba beszerkeszthetjük. Megszabadulunk tehát a bittérképek unalmas kódolásától.

Szöveges megjelenítéskor 27 különböző jelkészletet alkalmazhatunk, ezek különböző méretekben állnak rendelkezésünkre (6—24 képpontos magasság).

Bonyolult felhasználói programok fejlesztésekor jól használható az a



könyvtár, amely a különböző párbeszéd boxokat tartalmazza. (Ilyen lehet például az állományválasztó box.)

Érdekes további lehetőség a „virtual memory handler" (ezt a teljes verzió tartalmazza): átlépi az eddigi memóriakorlátokat, szükség esetén automatikusan használja virtuális memóriaként az EMS-t, illetve a merevlemezt.

Egy egyszerű „Task supervisor" a különböző folyamatok mintegy párhuzamos feldolgozását teszi lehetővé. Így folyamatosan megjeleníthetünk például órát az egyik ablakban.

A „Keyboard handler" segítségével megadott billentyűkhöz rendelhetünk egyes procedurákat.

Igazi különlegesség az animációmodul, amellyel az ikonokat mozgásba hozhatjuk.

A „Tegl Windows Toolkit II" fontos tartozéka mintegy 500 kilobájtos angol nyelvű dokumentáció és számos demóprogram. Mindent összevetve valószínűleg megnyeri a felhasználók nagy részének tetszését. A fejlesztési koncepció igen jól átgondoltnak tűnik. Látszik, hogy nagy kedvvel és a legapróbb részletekre is ügyelve készítették. A képernyők felépítésekor és ablakkezelésnél is megfelelő a sebesség. A shareware-verzió nem, de a teljes verzió tartalmazza a forráskódot, az EMM-támogatást, a virtuális memóriakezelést és a 27 jelkészletet. A Floppy.Lap júliusi száma mintaprogramokat közöl a Tegl Windows Toolkit II használatára, Turbo Pascal programozásnál.

V. P.

SR-Info adatbázis-fejlesztő rendszer

Mindazoknak ajánljuk az SR-Info behatóbb tanulmányozását, akik számára a többi jól ismert adatbázis-fejlesztő rendszer túlságosan drága lenne. A felhasználók nagyon gyorsan észreveszik, hogy szinte teljes értékű dBase-utánzattal dolgoznak. Kiterjedt utasításkészlete és számos makrója mellett integrált fordítót is tartalmaz. Az SR-Info-alkalmazásokat hívás előtt lefordítja, és csaknem ugyanolyan gyorsan futtatja, mint a C programok. A .DBF állományok mellett számos további adatbázisformátumot is kezel az SR-Info (ilyen például a .SDF). Egyidejűleg legfeljebb hat adatbázist nyithatunk meg. A programhoz tartozó bőséges — 800 kilobájtnál hosszabb — leírás, valamint az integrált help funkció biztosítja, hogy a felhasználó mindig hozzájusson a szükséges háttérinformációkhoz.

SolarSoft sikerlista

(Az 1991. áprilisi és májusi eladások alapján)

	No.	Programnév	Db	Programleírás
1.	421	PKZ110 & PKLITE & SHEZ	1	A „sűrítés" magasiskolája és Norton Commandere
2.	319	SCAN76 & NETSCAN	1	McAfee-féle vírusmegelőző, -detektor és -ölő
3.	470	MULTI-EDIT 5.0	1	A világon a legjobbnak tartott editor
4.	466	SKYGLOBE STAR GAZER	1	Mozgó, színes csillagtérkép!
5.	494	TEGLP WINDOWS TOOLK.	1	Ikongrafikus felület, ikoneditor TP-hez
6.	475	NEWSPACE	1	Harddiszkünk kapacitását megduplázza!
7.	096	AS-EASY-AS 4.00p	1	Lotus kompatibilis táblázatkezelő, egyszerűbb
8.	435	OPTIKS 2.18 & ICONVERT	1	PCX,PIC,GIF,TIF,GEM,MAC... grafikus konverterek
9.	485	BASIC COMPILER	1	Két ragyogó fordító — editorral!
10.	463	GAMES FOR WINDOWS	1	10+1 játék MS Windows 3.0 alá
11.	477	BACK & FORTH	1	Memóriamenedszer: 20 programot futtat egyszerre!
12.	M020	PC-JOG	2HD	Hatályos jogszabályok visszakereső rendszere
13.	383	4DOS V3.01a	1	COMMAND.COM pótló DOS héj: 50 új parancs
14.	474	JORJ POP-UP DICT.	2	58.000 szavas angol értelmező szótár
15.	480	GRASP 1.10C	1	Látványos animáció- és demókészítő program
16.	432	LZEXE & LIST 7.5e	1	Gyors EXE kompresszor, Vernon Buerg LIST PLUS-a
17.	425	POP-DBF 1.1 & dLite	1	Tárrezidens dBASE (EDIT/BROWSE/DISP stb.)
18.	304	TURBO TECHNO JOCKS	2	Szuper Turbo PASCAL unitok forrásaikkal!
19.	495	TEGLC WINDOWS TOOLKIT	1	Ikongrafikus felület, ikoneditor TC-hez
20.	472	SHARESPELL	1	Bővíthető, önálló helyesírás-korrektor
21.	484	SR-INFO	2	dBASE-kompatibilis fejlesztőrendszer!
22.	327	LHARC & LHICE	1	Japán szupertömörítő/önkicsomagoló program
23.	461	ZEPHYR 2.0	2	Komplett interaktív adatbázis-kezelő
24.	329	PC-MAGAZINE BENCH.	1	Az USA szaklap hardvertesztjei (v5.0)
25.	468	SUPER ASSEMB. EDITOR	1	TASM-ra kihegyezett programeditor
26.	406	XTAB	1	Statisztikai és grafikus csomag dBASE-hez!
27.	478	XTREE 2.0E	1	Az ismert, kisméretű, gyors fájlmenedzser
28.	422	FONTEDIT 5.7	1	HP lézernyomtatókhoz szoftfontok
29.	496	COMPLETE C C++	2	C++ kiterjesztés MSC/TC-hez!
30.	070	BLACK MAGIC	3	Grafikus módú hipertext, önálló futtató modullal

SolarSoft újdonságok, régebbi programok új verziói

(A programok angol nyelvűek)

No.	Programnév	Db	Típusa	Programleírás
018	AUTOMENU 4.7	1	DOS-keret	Ügyes DOS-menügenerátor
026	NEWKEY 5.4	1	Bill. makró	Billentyűzetdefiniáló
047	TELIX 3.15	1	Komm.	Programozható, ZMODEM
054	PC-WRITE 3.03	2	Szövegszerk.	Szövegfeldolgozó program
059	GALAXY WORD 3.01	1	Szövegszerk.	WS-komp., menüvezérelt, ékezetes
091	PC-FILE+ 5.0	3	Adatbáziskez.	A legjobb shareware adatbázis-kez.
096	AS-EASY-AS 4.0 & PIVOT	1	Táblázatkez.	Lotus-komp., de még jobb, grafika
186	IDCSHELL & NARC 4.0	1	Adattömörítő	A NARC ikertestvére, ZIP-et is tud
207	PC-CALC+ 2.0	2	Táblázatkez.	Listavezető táblázatkezelő
215	EXPRESS CALC 4.1	1	Táblázatkez.	Lotus 1-2-3-klón
299	INSTACALC 2.52	1	Táblázatkez.	Tárrezidens Lotus
319	SCAN 7.2C76 & NETSCAN	1	Vírusvédelem	1991. áprilisi (McAfee)
435	OPTIKS 2.18 & ICONVERT	1	Grafikus	PCX,PIC,GIF,TIF,GEM... konverter
497	BGI TOOLKIT	1	Fejlesztő	Borland BGI fejl. készl., fonteditor
504	PC-TYPE + 2.0	3	Szövegszerk.	Szuper program, grafikát kezel
505	GALAXY LITE 1.6	1	Szövegszerk.	A nagysikerű Galaxy felújítása
506	PROLITE 1.01	1	Szövegszerk.	A PC-Write egyszerűsített változata
507	PC-BROWSE&NG_MAKER	1	Hipertext	NG decomp. és szenzációs NG-klón
508	PROCUBE LITE 1.0	1	Táblázatkez.	A Qubecalc 3D továbbfejl. változata
509	ZIPVIEW & LHA 2.10	1	Adattömörítő	Norton kieg. A PKZIP-nél jobb
510	ARJ 2.0	1	Adattömörítő	PKZIP/LHA-nál is hatékonyabb

Jön, jön, jön... és már itt is van!

A #018-as számú **Automenu 4.7** már megegyezik a kereskedelmi változattal. Jelentős továbbfejlődésen az MDF editor esett keresztül. (Semmi köze a kormánypárthoz, a rövidítés itt a Menu Definiton File helyett áll!)

Az #054-es sorszámú **PC-Write 3.03** programot sokak kívánságára szereztük be. (A Cédrusnál már többen is ezt a szövegszerkesztőt használják.) A program automatikus elválasztáskezelése magyar viszonyokra is kielégítőnek mondható. Egyszerre többhasábos dokumentum is szerkeszthető. A program egyszerűsített (és csak egy lemezből álló, nem programozható) változata is elérhető **Prolite** néven az #506-os számon.

Az #059-es **Galaxy Word 3.01** parancsszinten WordStar-kompatibilis, egérrel is vezérelhető szövegszerkesztő. A 3.0-ás változattal szemben ez már hibátlanul kezeli a 128 feletti ASCII kódú karaktereket is, így aki ezt a lemezt megvásárolja, teljes értékű editorhoz jut. A Galaxy Word korábbi változatát újra megjelentették a SolarSoft kínálatában #505-ös számmal **Galaxy Lite 1.6** néven, ez magában foglal egy angol nyelvű helyesírás-javítót is.

A #091-es számú **PC-File+ 5.0** adatbázis-kezelő rendszer a maga kategóriájában világszerte legnagyobb példányszámban eladott és használt szoftver, dBASE III-kompatibilis adatbázisokon dolgozik, könnyen elsajátítható nyelven programozható is.

A #186-os lemezen található **IDC-Shell** nevű DOS-keretrendszer PKZIP 1.1-kompatibilis archív állományokat is képes létrehozni és kezelni. A NARC-kal adják párban, mely ZIP és ARC állományok tartalmát jeleníti meg szemléletesen. Használatuk a Shezzel ellentétben nem feltételezi a PKZIP, PKUNZIP, PKPAK, PKUNPAK adattömörítő programok meglétét. Kezelésük kényelmes.

A táblázatkezelő szoftverekben járatos ínyencek számára az ismertebb shareware táblázatkezelők legfrissebb változatát kínálja a SolarSoft: **PC-Calc+ 2.0** (#207), **Express Calc 4.1** (#215) és a memóriarezidens **Instacalc 2.52** (#299). Egy valódi háromdimenziós táblázatkezelőt is ajánlhatunk, a **Procube Lite**-ot (#508), mely a népszerű Qubecalc 3D szerves folytatása, egy 256x256x256-os méretű kockát valósít meg. Az egymás alá képzelt táblázatokat a kockának egy másik lapjára merőleges irányból szemlélve olyan érdekes kimutatásokhoz jutunk, melyet egy hagyományos táblázatkezelővel magunknak kellett volna megtervezni. A program használata megszabadít a képernyőn megjeleníthető kvázi háromdimenziós ábrázolás gondolati bilincseitől. Innen már csak egy lépés a négydimenziós táblázatkezelő, amelyet csak elméleti matematikusok és fizikusok tudnak majd kezelni kettős síkproblémák megoldására?!

Az editorok sora újabb tekintélyes példánnyal bővül, a grafikát is kezelni tudó **PC-Type+ 2.0**-val (#504). Az #507-es **PC-Browse** egy interaktív hipertext rendszer, melyet leginkább a közkedvelt Norton Guides-hoz tudnánk hasonlítani. Ez egeret is kezel. A lemez meglepetésként tartalmaz egy NG visszafordítót is!

Az adattömörítés szerelmesei mindjárt három csemegével gyarapíthatják arzenáljukat. A **Zipview** program a Norton Commander 3.0 zseniális kiegészítése, az összes ismert archív formátumot felismeri, tartalmát megjeleníti, képes ki is csomagolni. A zökkenőmentes beépítéshez egy átírt NC-MAIN.EXE is található a lemezen. Az LHARC 1.13C legújabb változata, a PKZIP 1.1-gyel összemérhető sebességű, de hatékonyabban zsugorító **LHA 2.10** is az 509-es lemezen található. Az #510-es **ARJ 2.0** viszont még az LHA-n is túltesz. Jobban tömörít, azonos gyorsaságú. Backup készítésére is kiválóan alkalmas, hiszen ha floppyra mentenénk, és az összecsomagolt állomány nem férne el egyben az adott lemezen, az ARJ nem esik kétségbe, kéri a következő lemezt, s arra folytatja a mentést. A csomag tartalmazza az ARJ-formátum magyarázatokkal ellátott szerkezetét, fejlécét.

Figyelem! Az ARJ a PKZIP-pel ellentétben nem shareware, hanem freeware, ami azt jelenti, hogy pénzért forgalmazott, profitorientált kereskedelmi termékeinkhez az ARJ-t felhasználhatjuk (például az ARJ önkicsomagolót). Lemezeket takaríthatunk meg egy-egy nagyobb adatbázis-kezelő rendszer ARJ-vel történő installálása esetén.

Híreink szerint Phil Katz és csapata, a PKWare összeállt az IDC-vel (Infinity Design System, lásd följebb IDCShell és NARC) és egy új adattömörítő szabványt dolgoznak ki. Lehet, hogy máris elkéstek?

Mindenesetre nem kell nagy jóstehetség ahhoz, hogy bátran ki merjük jelenteni, az ARJ-t nemsokára a SolarSoft népszerűségi lista felső régióiban fogjuk találni.

Herczeg József

Íme az erő: PowerBasic 2.10

Sok programozó érezhette becsapottnak magát, amikor észrevette, hogy nem fejlesztik tovább a széles körben elterjedt Turbo Basicet. A kölni Orgatec jelentett fordulatot számukra, itt mutatták be ugyanis a Kirschbaum Software cég új termékét, a PowerBasic-et. Ez a mintegy 40%-kal gyorsabb PowerBasic amellett, hogy igen kényelmesen vezeti a felhasználókat munkájuk során, számos új lehetőséget is tartalmaz: unitok és objektumok linkelését, a hálózati támogatást, nyomkövetést mind forrás-, mind pedig alacsony szintű kódok esetére.

Újdonság még, hogy már a 60 kilobájtnál hosszabb stringek is tárolhatók, 80 bites BCD lebegőpontos műveletekkel dolgozhatunk, valamint az, hogy megjelent a 80x87 procedúra-emulátorcsomag, és mód van az EXE-kód optimalizálására is. Mindezeken kívül számos új utasítás, függvény, új adattípus teszi kényelmessé a munkát. További segítséget nyújt a 170 rutint tartalmazó PowerTools csomag, ennek felhasználásával rengeteg időt és energiát takaríthatunk meg programfejlesztés során.

A DOS Shareware nyomán (1991/3.)

Minél rosszabb — annál jobb!

Az Alaplapban idén már többször is foglalkoztunk azzal, hogy milyen nehézségek tornyosulnak egy magyar „spelling checker" megalkotása előtt, és jeleztük, készülnek a programok, csak azt nem tudjuk, mikor lesznek készen. A budapesti Ifabón azután egyszerre két program is megjelent: a „NyelvÉsz" és a „Helyes-e". Kíváncsian vártuk, mennyire felelnek meg várakozásunknak.

Otthonosan mozogván a magyar nyelv feldolgozásában, kapásból tudunk mondani 30-40 nehezen kezelhető problémát, és igazi meglepetést okozott volna, ha e programok mindegyiket jól kezelnék. De a dolog használhatósága mégsem ezen múlik, hanem a gyakoriságon, mert lehet, hogy egy fogós nyelvi probléma a használat során 100 évenként egyszer fordul csak elő. Ezért azt a megoldást választottuk, hogy 5 különböző forrásból mintegy 100 kilobájtnyi nyers (lektorálatlan) szöveget szereztünk be, és ezen próbáltuk ki a két programot.

E cikk címe lehet, hogy egy kicsit furcsa, de nem a programok iránti ellenszenvet fejezi ki, pusztán arra a tényre utal, hogy vannak, akik tudnak helyesen írni, illetve gépelni, és vannak, akik nem. (Sajnos magam is az utóbbi kategóriába tartozom.) Összefoglalóan megállapítható: a két program jelenleg olyan állapotban van, hogy az első csoportba tartozóknak inkább bosszúságot okoznak, míg aki a második csoportba tartozik, az talán a jelenlegieket is — de a következő verziót mindenképpen — használni tudja.

Mindkét programnak számos üzemmódja van (erre később még kitérünk), de praktikus értékét nézve csak egy használható: az, amikor „batch" üzemmódban beolvas egy fájlt, és egy hibalistát ad ki arról, hogy milyen ismeretlen szavakat talált és hányadik sorban.

A listát három részre lehet osztani: a valóban hibás szavak listájára, a program által hibásnak vélt, de valójában helyes szavak listájára, végül azon szavak listájára, melyeket „engedéllyel" nem ismer a program: ezek a tulajdonnevek, a rövidítések és az idegen szavak.

A használhatóság az arányokon múlik. Ha a fel nem ismert szavak teljes listáján minden második szó valóban hibás, akkor a programot jól tudjuk használni. Ha csak minden huszadik, akkor nemcsak bosszankodunk, de fellép a „fától nem látni az erdőt" effektus is, és elég nehéz megtalálni a valódi hibákat.

A másik érdekes kérdés, hogy milyen hibás szavakat nem talál meg a program. A két hibalista összevetése elég érdekes eredményre vezetett. Sommásan azt mondhatjuk: a „Helyes-e" mintegy 25-30%-kal több hibát fogott meg, de ennél is nagyobb mértékben nézett helyes szavakat helytelennek. Megpróbáltunk keresni a szövegben olyan hibát is, amelyet egyik program sem vett észre. Nem sikerült, pedig talán volt ilyen. Olyan szó persze akadt jónéhány, amelyik helyes volt, de egyik program sem fogadta el annak.

A „betáplált" szövegek az alábbiak voltak:

I. Egy önéletrajz.

II. A 3C 274 kvazár — cikk a Tudomány című lapból.

III. Hardveres vírusvédelem — cikk az Alaplapból.

IV. Romániai zsidóüldözésről szóló novella.

V. Interjú az igazságügyi államtitkárral — a Ring című lapból.

Hogyan összegezhetjük tehát a tapasztalatokat? A magyar nyelvben a ragozás igen bonyolult, és azt egyiknek sem sikerült tökéletesen beprogramoznia. A „NyelvÉsz" kevesebb hibát vét, de azért még van mit javítani rajta. A hibák rendszerint a szóösszetételek nem elég kifinomult kezeléséből származnak. A „Helyes-e" kissé kidolgozatlannak tűnik, sok helyes szót néz hibásnak, de a hibákat többnyire észreveszi.

A problémák a ragozott névmásoknál, a középfoknál és az „-i" képző körül csoportosulnak (engemet, valahová, lassabb). A programoknak kezelniük kellene az „össz-" előtagot és a „-féle", „-fajta", „-szerű" utótagot is.

Az alapszótár tekintetében mindkét program azt állítja, hogy az Értelmező Kéziszótárból indul ki, de vannak benne kisebb lyukak (csöppet, csúfot, fülik stb.). Mivel ezek a szavak választékosak, egy igen terjedelmes cikkben sincs több belőlük, mint három-négy.

Az összehasonlító teszt eredménye

(Törtvonal előtt a „NyelvÉsz", mögötte a „Helyes-e" megfelelő adata.)

	I.	II.	III.	IV.	V.
Terjedelem (kbájt)	28	35	21	17	11
Terjedelem (sor)	416	614	420	297	168
Terjedelem (szó)	3045	4618	2739	2306	1447
Ismeretlen szavak száma	194/226	309/417	240/275	138/152	26/38
Megtalált hiba	85/107	12/17	73/78	4/5	4/5
Meg nem talált hiba	22/1	5/0	7/2	1/0	1/0
Rosszul elemzett jó szó	21/41	37/60	10/27	23/35	12/25
Jogosan fel nem ismert szó	96/78	260/340*	157/170	111/112	10/8

* — A „Helyes-e" a „3C 274" szót, amely egy kvazár neve, kb. 70-szer kifogásolta. A „Nyelvész" nem szólt rá semmit.

A tulajdonnevek kezelését tekintve a két program koncepciójában különbözik. A „NyelvÉsz"-ben egyetlen tulajdonnév sincs, de mindegyiket megugatja. A „Helyes-e" viszont tartalmaz egy sor keresztnevet és helységnevet, ami az indokolatlan hibák számát jelentősen csökkenti. (Az ember ugyan elgondolkozik rajta, hogy miért ismeri Bácskát, a Bánátot pedig nem; de világos, hogy valahol meg kell húzni a határt.)

A rövidítések közül egyik program sem tudja kezelni a legismertebbeket sem (MÁV, OTP, tsz stb.). Mivel ezeket többnyire kötőjellel toldalékoljuk, alkalmazásuk nem tűnik nagy művészetnek. A kötőjellel egyébként is gond van. Kezelését egyik program sem szereti, még olyan rendkívül gyakori esetekben sem, mint az 1848-as, 56-os, 44-ben stb. Nem is szólva a 6:3-as szabályról, amit elég könnyű lenne felismerni. (A hat szótagnál hosszabb, legalább három szóból összetett szavakat főösszetevőjüknél kötőjellel kell tagolni.)

A vizsgált programokban két neuralgikus pont van: az összetett szavak és az idegen szavak. A mai magyar nyelv tele van idegen szóval. Az értelmező kéziszótár a divatosak közül viszonylag keveset tartalmaz, hiszen elég régen is készítették. Pedig egy sor szó nemigen tekinthető már idegennek, sokukat magyar ékezetes betűkkel is írjuk (média, informatika, funkcionális, paraméter, vírus stb.). Ebben a tekintetben mindkét program elég szegényes, egy-két ezer szót beléjük kellene tölteni. Ezt magára az alapszótárra értem. Az persze más kérdés, hogy bizonyos szövegekhez kiegészítő szakszótárra is szükség van.

Az összetett szavak írása még a helyesírási szabályzat szerint is „olykor határozatlannak látszik". Noha vagy fél tucat szabály van rá, gyakran nem világos, hogy melyiket kell éppen alkalmazni, és szinte minden különíró szabály esetén ott áll, hogy „a kialakult szokást megtartva egybeírjuk a következő kapcsolatokat..."

Vegyünk két egyszerű példát a kvázáros cikkből: „nagyenergiájú (elektron)" és „gázfelhő". Az elsőre a következő szabály vonatkozik: „Ha két különírt szóból álló minőségjelzős kapcsolat második tagjához -i, -beli, -s, -ú, -nyi képző járul, a különírást megtartjuk." Hasonlóképpen nyilatkozik a Műszaki Helyesírási Tanácsadó Szótár is. Ezzel szemben a Természettudományos Kisenciklopédia és más fizikakönyvek egybeírva használják. A második példára a következő szabályok tartoznak: „A minőségjelzős kapcsolatok tagjait általában különírjuk." ... „Az anyagnévi jelzőt, ha egyszerű szó, egybeírjuk a nem összetett főnevekkel." ... „Az úgynevezett jelentéstömörítő összetételeket (gázból való felhő) mindig egybeírjuk." Na már most, mindenki eldöntheti magában, hogy egy gázfelhőben a gáz és a felhő milyen kapcsolatban van egymással.

Látható tehát, hogy az egybe- és különírás nem egyszerű probléma, a két program másként is döntött. A „NyelvÉsz" program fejlesztői először úgy gondolták, hogy szinte bármilyen szókombinációt hagynak egybeírni. Ebből azután szörnyű hibák keletkeztek. Nemcsak a hibás egybeírásokat hagyja megjegyzés nélkül a program („olymódon" — „oly módon" helyett, „résztvesz" — „részt vesz" helyett stb.), hanem egy csomó hibás szót intarziázott össze kisebb szavakból („kin|levez|ék" — „kinevezték" helyett, „út|ól|só" — „utolsó" helyett stb.). Ennek kompenzálására bevezettek egy opciót, hogy hány szóig lehet összetenni szavakat. De persze ez sem jó megoldás, hiszen így –c1 opciót adva a program minden összetett szót hibásnak tekint, pedig minden szöveg tele van összetett szavakkal.

Most már a program készítői is belátják, hogy új megoldást kell keresniük az összetett szavak kezeléséhez, Seregy Lajos meg is ígérte, hogy az őszre vagy télre elkészülő új verzió már tartalmazni fogja ezt a bővítést. Hogy a probléma nem teljesen megoldhatatlan, az abból is látszik, hogy a „Helyes-e" félig-meddig megoldotta.

Nézzük meg most a két programot mint programot.

Mindkét program rendelkezik egy nagy pozitívummal, egy leíró fájlból vezérelhető, hogy milyen kódot használjon. Az alapeset mindkét leírás szerint a CWI szerinti kódkiosztás.

Amikor az első szövegre ráengedtem a „NyelvÉsz"-t, majdnem 600 hibajelzést kaptam, és rengeteg töredék szó volt közte. Néhány perc tűnődés után rájöttem, hogy egyetlen „ő"-t és „ű"-t sem ismert fel a program. Ekkor a kódleíró fájlhoz fordultam, és sejtelmem nem csalt, nem a CWI-kód volt benne leírva. Majdnem két percig tartott, amíg kijavítottam, és a hibák száma máris 300-zal csökkent.

A „Helyes-e" programnak három üzemmódja van:

— Az alapüzemmódban beolvas egy szöveget és hibajegyzéket készít.

— Szótározó üzemmódban a felhasználó begépelhet egy szót, és a program vagy azt mondja, hogy megvan, vagy azt, hogy nincs. Egy kicsit lehet vele játszani, de az ember hamar megunja.

— Interaktív javító üzemmódban a program beolvas egy fájlt, a hibáknál megáll, kiírja a hibás szót, beolvas helyette a klaviatúráról egy másikat, azt is ellenőrzi, és a javított szöveget fájlba írja. Első pillanatra ez a megoldás elég jónak látszik, csak a gyakorlatban derül ki, hogy nem az. Az esetek nagy részében ugyanis látjuk, hogy a szó rossz, de nem tudjuk a szövegkörnyezet nélkül, hogy mi lenne a helyes. Legalább 5 szót ki kellene írni előtte és utána.

A „NyelvÉsz" alapműködése, hogy klaviatúráról olvas és képernyőre ír. Mivel az MS-DOS lehetővé teszi, hogy a standard inputot és outputot átirányítsuk, így az inputját veheti egy fájlból is, és teheti az outputot egy fájlba is. Ilyenkor persze a képernyő elsötétül, a gép semmi életjelet nem mutat, egy lassú gép és nagy fájl esetén akár 20 percig se, és csak Ctrl-Alt-Del-lel lehet megállítani. Nem túl megnyugtató megoldás. Legalább 100 vagy 500 szó után küldhetne egy apró csillagot a képernyőre, vagy valami hasonlót: „Még élek, csak lassan dolgozom."

A programnak számos opciót lehet és kell is megadni.

–s. Csak a hibás szót és a helyét írja ki. (Különben a jó szavakra is kiírja, hogy megvan.)

–h. Kiírja, hogy a szó milyen részekből áll és hol lehet elválasztani. (Ebből lehet látni, hogy a hibás szót milyen részekből állítja össze.)

–c. A szóösszetétel hosszát szabályozza.

–d. Felsorolja, hogy egy (esetleg több) betű kihagyásával, beszúrásával, átírásával milyen (jó) szavakat kaphatunk a (rossz) szóból. Két sajnálatos mellékhatása volt az én programomban: nemcsak a rossz, hanem a jó szavakra is ad tanácsot, az –s sem kapcsolja ki. Így minden fájlt legalább 10-szeresére duzzaszt. A szóösszetételt egyre korlátozza, és így egész más hibalistát ad, mint az eredeti program.

Mindkét program helyfoglalása viszonylag kicsi. A „Helyes-e" futási ideje is egészen kitűnő, remélhetőleg a javítások, továbbfejlesztések sem csökkentik a sebességét. Ehhez képest a „NyelvÉsz" kissé lassú, 25 MHz-es gépen még kellemes, de egy 8 MHz-es XT-n bizony már idegesítő.

Magam részéről remélem, hogy őszre mindkét program újabb változattal áll elő, és az itt említett problémák jelentős része megszűnik.

Farkas Ernő

Framework-world

Tiszta Fred, a kapitány

A Framework programról júniusi számunkban a táblázatkezelők között már olvashattak. Az integrált programokat (a Framework is ilyen) elsősorban ügyviteli alkalmazásokban használjuk. Egy komplex ügyviteli rendszer kényelmes kezeléséhez azonban ma már szükséges egy barátságos felhasználói felület, amely segít az adatok bevitelében, a különböző listák lekérdezésében, tájékoztat arról, hogy éppen mi történik, eligazít a tennivalók között. A Framework-beli munka körvonalai adottak: szükség van a megfelelő táblázat-, adatbázis- vagy szövegkeretekre stb. Az adatok beviteléhez, lekérdezéséhez, a különféle listák elkészítéséhez a Framework programozási nyelve, a Fred segítségével barátságos felhasználói rendszert készíthetünk. Egy ilyen program természetesen magyar nyelven kommunikál a felhasználóval. Ötrészes sorozatunkban bemutatjuk a Fred világát. Ismertetjük a nyelv elemeit, a programírás szabályait, a különböző jellegzetes funkciócsoportok alkalmazási lehetőségeit (például tartománykezelés, függvénykészítés, makrókészítés, billentyűzetkezelés, vezérlésátadás stb.); s bemutatunk néhány programozási trükköt is.

Feltételezzük a Framework III interaktív használatának ismeretét. Terjedelmi okokból természetesen nem térhetünk ki minden beépített függvénynek az összes paraméterre kiterjedő részletes magyarázatára. Ezt bárki megtalálja a Framework III dokumentációjában vagy (kevésbé részletesen) Bill Harrison magyarul is megjelent Bevezetés a Framework III használatába (Novotrade, 1991) című könyvében. Mivel a Framework III-ban programozási szempontból csak kevés az eltérés a korábbi változathoz képest, még Barakonyi Károly: A Framework II használata kezdőknek (LSI ATSZ, 1989) című könyve is használható.

Első alkalommal mindazokat a lehetőségeket mutatjuk be, amelyekkel gyorsabban érhetünk célhoz: a rövidítések, makrók készítését, használatát. Szó lesz a library szerepéről, ismertetjük a Fred általános szabályait, s elkészítjük az első kis programokat is.

Minden cikk végén „házi feladatokat" adunk, amelyek megoldását várjuk olvasóinktól.

Eszközök a gyorsabb munkához

A Library

A Framework betöltése után automatikusan betöltődik egy különleges fájl is: a LIBRARY.FW3. Ez egy konténerkeret, amelyben hasznos segédeszközöket találunk: makró- és rövidítésdefiníciókat, gyakran szükséges táblázatokat, programokat, saját függvénydefiníciókat. Ezek mindig „kéznél vannak", így a Framework III-mal végzett munka során bármikor használhatók.

Cikksorozatunkban gyakran hivatkozunk egyes elemeire: természetesen ezek az eredeti, Magyarországon a Novotrade által forgalmazott programcsomagban levő Library-keretben megtalálhatók.

Rövidítések

Különböző dokumentumok: termékismertetők, üzleti levelezés, szerződések elkészítésében hasznos lehetőség a rövidítések használata, ha például Magyar Tudományos Akadémia helyett elég mta-t begépelni, s egy billentyűkombináció (**Alt-Backspace**) leütésére a rövidítés kicserélődik a teljes kifejezésre.

Ehhez a **CREATE** menü **Macro/Abbreviations** parancsának indítása után először a rövidítést kell begépelni, majd egy entert követően az egész kifejezést, amit Ctrl-Break-kel kell lezárni. A művelet során a Libraryben az Abbreviations szekcióban létrejön egy szövegkeret, amelynek belsejében a teljes kifejezés áll, a keret neve pedig a rövidítés. Ha a későbbiekben is szükségünk lesz erre a lehetőségre, mentsük el a Library-keretet (válasszuk ki, és Ctrl-Enter).

Tovább gyorsíthatjuk a munkát, ha a rövidítések használatához bekapcsoljuk a **WORDS** menü **Word Options** almenü **Expand Abbreviations** kapcsolóját. Ekkor ugyanis a rövidítés helyére akkor is bekerül a teljes kifejezés, ha szóközt, vesszőt vagy más szóvégkaraktert ütünk le.

Az eredeti Framework III Libraryben három, számunkra kevéssé használható rövidítésdefiníció van: AT — Ashton-Tate, GB — Great Britain, UK — United Kingdom.

Ezeket a saját munkánkhoz szükséges, gyakrabban használatos rövidítésekkel kicserélhetjük (például: bme — Budapesti Műszaki Egyetem, ksh — Központi Statisztikai Hivatal, fw3 — Framework III).

Makrók

Makrón olyan építőelemet értünk, amelyet a munkánk során mint speciális funkciót használhatunk. Megvalósításuk (megvalósíthatóságuk) alapján kétféle makróról beszélünk: **billentyűmakróról**, amely billentyűleütés-sorozatot jelent; valamint a **programmakróról**; ez utóbbi egy olyan program, amelynek eredménye rendszerint egy adat (napi dátum valamilyen kü-

lönleges formátumban, napi árfolyamérték stb.).

A makródefiníció rendszerint a Libraryben kerül tárolásra. A definiáló keret neve a makrót hívó alt-? billentyűkombináció. (Felhasználhatók a betűk, számok és a funkcióbillentyűk.)

Billentyűmakró

Ha gyakori billentyűsorozatot makróként definiálunk, akkor hívása egyetlen **alt-?** billentyűkombinációval történik. A definiáló keretben a szükséges billentyűnevek találhatók. Billentyűmakró létrehozható menüparanccsal — **CREATE Macro/Abbreviation** — (ekkor a definiáló keret automatikusan létrejön a Libraryben, a Macros szekcióban), de úgy is, hogy a Libraryben létrehozunk egy üres szövegkeretet, s ebbe írjuk a makró definícióját (az egymás után leütendő billentyűk neveit). A billentyűneveket a Framework III helpjében megtaláljuk. Általános szabály, hogy a billentyűkombinációkat, különleges billentyűk nevét kapcsos zárójelbe ({ }) kell tenni.

Az eredeti Framework III Libraryben két billentyűmakró található: az Alt-F1 és Alt-F2.

{ALT-F1}:
{CTRL-W}L0{Return}{Ins}
P0{Return}{Ins}A

Ez a makró 0-ra állítja a bal margót és a bekezdés beljebbkezdését, azaz a bekezdés valamennyi sora a keret szélén, a bal margónál kezdődik. A szövegszerkesztésnél az általános bekezdéstípusra való visszaállásnál használjuk, például egy középre vagy beljebb igazított cím beírása után.

{ALT-F2}:
{End}{Ctrl-C}E{Ctrl-Uparrow}
{F6}{Home}{F7}{End}{Return}
{Uparrow}

Ez a makró a konténer- vagy vázlatkeretben az aktuális szinten egy első üres keretet hoz létre. Ehhez először az Enddel az aktuális szekció utolsó keretére lép, készít egy üres keretet, amely így utolsóként jön létre. A felfelé mutató nyílbillentyű a korábban utolsó keretet választja ki, majd F6-tal a kiválasztást kiterjeszti a szekció első keretéig. Az így kijelölt kereteket a szekció végére mozgatja (a létrehozott új keret után), majd a kurzort egy kerettel feljebb lépteti (a létrehozott új keretre).

Programmakró

Egyszerű adatfeldolgozási rutint is definiálhatunk makróként, ennek a hívása is egy alt-? billentyűkombinációval történhet. Hogy mindig kéznél legyen, célszerű a Libraryban elhelyezni a makrót definiáló keretet. A programot ennek az üres szövegkeretnek a szegélyére írjuk (a képletterületre), s a keret neve a hívó alt-? billentyűkombináció.

Az eredeti Framework III Library programmakrókat tartalmaz az Alt-F3, Alt-F4, ..., Alt-F10 kulcsok alatt.

Csak egy-egy alkalmazáshoz tartozó makrókat nem tárolunk a Libraryben, mert akkor nagyon megnőne a mérete, s csökkenne az aktuális feladatunk elvégzéséhez rendelkezésre álló szabad memóriaterület.

Ilyenkor legalább két keret szükséges: a makrót definiáló, tetszés szerinti elnevezésű keret, valamint az, amely az előbbit egy alt-? makróhíváshoz rendeli. Célszerű ezt úgy megoldani, hogy az összerendelő keret belsejében helyezzük el a makrót definiáló keretet. A külső keretre írjuk a **@setmacro({alt-?},keretnév)** függvényt, ahol második paraméternek írjuk a belső keret nevét. Ha ezt a függvényt végrehajtjuk, a továbbiakban a megadott alt-? billentyűkombinációval a belső keretben definiált makró hívható.

Fred-programok

A Framework programozott használatának az a célja, hogy bizonyos műveletek elvégzését automatizáljuk. Program segítségével például az adatfeldolgozási folyamat (vagy annak egy része) vagy akár a menüparancsok egymásutánja is indítható egyetlen paranccsal. Ez legegyszerűbben a makróként definiálással történhet, de akár komplett felhasználói rendszert is készíthetünk.

A Framework III programozási nyelve a Fred.

Fogalmak, szabályok a programíráshoz

A Framework programozott használata megfogalmazás utal arra, hogy a Fred-programok futtatásához szükség van a Framework háttérre, nem hozható létre közvetlenül futtatható (.COM vagy .EXE) program. Ez azonban egyúttal azt is jelenti, hogy a Framework minden, interaktív módon hozzáférhető lehetősége a Fred nyelvű programokban is rendelkezésre áll.

Képletekkel, egy-egy adat kiszámításához szükséges „utasításokkal" az interaktív használat során a táblázat- és az adatbázis-kezelésben találkozhatunk. Ezeket a megfelelő táblázatcellában vagy adatbázismezőben a képletterületre kell beírni.

Programot is képletterületre írunk, méghozzá egy üres szövegkeret szegélyének a képletterületére. A képletterületre az F2 leütését követően írhatunk.

Tekintsük át, milyen elnevezéseket használunk a Fred nyelvű programkészítés során!

Kifejezés

A programot alkotó legkisebb értelmes egység, amely lehet egy operátorokkal (műveleti jellel vagy jelekkel) kijelölt művelet vagy függvény. A függvények paramétereként, illetve az operátorokkal megadott műveleteknél argumentumként konstans értékek vagy hivatkozások (táblázatcella, adatbázismező vagy keret hivatkozás) szerepelnek.

Példák:
C3+7, 1991-NEVSOR.Munkábalépés,
@SUM(NEVSOR.Fizetés),
@AVG(TABLA.B3:Tabla.C10,0).

Formula

Egyetlen képletterületre írt kifejezés vagy kifejezések összessége. Ha a formula több kifejezésből áll, a kifejezéseket vesszővel kell egymástól elválasztani. Az utolsó kifejezés végére azonban nem szabad vesszőt tenni. A képletterületet formulaterületnek is nevezik. Az e cikkünk végén tárgyalt egyszerű programok formulák.

Program

Egy program állhat egyetlen formulából, de felhasználhat másik formulában már elkészített programmodulokat is. A műveletek eredményének tárolására, az értékek átadására szolgálnak a változók, amelyek elnevezésére nincsenek megkötések, állhatnak betűkből, számokból, hosszuk gyakorlatilag korlátlan (max. 64 000 karakter). A programban betöltött szerepük szerint megkülönböztetünk lokális és globális változókat.

Változótípusok

Lokális változó csak egyetlen formulában használható, formulák közötti értékátadásra nem. A lokális változókat a formula elején (de mindenképpen az első hivatkozás előtt) deklarálni kell, a **@LOCAL( )** függvény argumentumaként. Értelemszerűen a változóneveknek a formulán belül egyedieknek kell lenniük. Például: @LOCAL(str,i,k).

Globális változókat használunk formulák közötti értékátadásra. Globális változót a formulákban külön deklarálni nem kell, hanem létre kell hozni egy üres szövegkeretet, melyet a globális változó nevével kell elnevezni.

Ha egy formulán belül ugyanazzal az elnevezéssel lokális és globális változót is használnánk, a hivatkozás lokális változóként értelmeződik.

Értéktípusok

A Fred programban minden kifejezésnek van értéke (ami nem feltétlenül numerikus, lehet karakteres, dátum, logikai, sőt Fred-konstans). Ez a tulajdonság a szokásos matematikai értelemben vett függvényjellemzőkkel megegyezik (függvény — függvényérték).

A Fred belső függvényei

A Fred mintegy 170 beépített függvénnyel rendelkezik. Vannak matematikai, trigonometrikus, statisztikai, pénzügyi stb. függvények, amelyek segítségével például táblázatok adatait határozzuk meg. Más függvények azonban a programok elkészítéséhez szükségesek (makrókészítő függvények, vezérlési függvények, állapotérzékelő függvények stb.).

A függvények nevét @ előzi meg, a függvény argumentumait ( ) közé kell tenni, egymástól vesszővel (,) elválasztva.

Egyszerű programok

Egyszerű programoknak nevezzük azokat, amelyek egyetlen formulából állnak, különleges vezérlési szerkezetet (programelágazás) nem tartalmaznak. Ezeken mutatjuk be a Fred-programok legfontosabb tulajdonságait, s a legalapvetőbb függvényeket (adatbevitel, eredmény kijelzése, változók használata).

Adatbevitel billentyűzetről (@inputline)

A függvény segítségével karakteres típusú (string) adat bevitele lehetséges. Az adatbevitelt Enter zárja le. Más típusú adat a konvertáló függvényekkel állítható ebből elő.

Az @inputline függvényt rendszerint értékadó utasítás jobb oldalaként használjuk.

Például: @inputline("Írja be az évszámot!","1991",,#YES).

Ez a kifejezés a képernyő alján középre kiírja az **Írja be az évszámot!** szöveget, a szerkesztősor bal szélén megjeleníti az **1991** default stringet mint szokásos értéket. Ha ez megfelelő, Entert ütve az 1991-et visszük be, különben az első beírt karakter eltünteti a default adatot.

Adatkivitel: az eredmény megjelenítése a képernyő alján (@prompt)

Gyakran fordul elő, hogy a program eredményét a képernyő alján közöljük a felhasználóval. Az üzenet az üzenetsor bal oldalán jelenik meg, de elhelyezhető középen is. Mivel a program előző kifejezései hagyhattak valamit az üzenetsorban, a @prompt használata előtt célszerű elvégezni az üzenetsor törlését az **@eraseprompt** függvénnyel.

Vigyázat! Az @echót be kell kapcsolni (ha kikapcsoltuk korábban a programban) a @prompt előtt, mert különben nem jelenik meg az üzenet! Az @echo(#off) a képernyőváltások letiltását írja elő a következő @echo(#on)-ig.

A Framework III programmakrók között két egyszerű programot tartalmazó van: Alt-F3 és Alt-F5.

{ALT-F3}:

```
; Types at cursor position
@performkeys(@date4(@today))
```

Ez a program egyetlen kifejezésből áll, amely a kurzor aktuális pozíciójába beírja a gépi dátumot @date4 formátumban. Például: 26th February 1991.

Ezt a makrót a hazai viszonyok között érdemes lecserélni a magyar helyesírási szabályoknak megfelelő dátumformára.

{ALT-F5}:

```
@prompt("you hare = " &
@business(@memavail, 0)
& " more bytes of memory
available.", 17)
```

A makró az üzenetsorba egy szöveges üzenet részeként kiírja a szabad RAM-memória méretét (a számot business formátumban), a sor bal szélétől 17 karakterrel beljebb. A @business függvény egy konvertáló függvény: a számból business formátumú stringet állít elő.

A továbbiakban a program első sorában szereplő komment sort (bármely sorban a ; után a sor végéig beírt szöveget tekinti magyarázatnak a rendszer) azért helyeztük el, mert itt adjuk meg a programot tartalmazó keret nevét. Nézzünk néhány példát egyszerű programokra:

1. példa. A program billentyűzetről vár egy nevet, majd az eredmény-string a név hosszát is megadja:

```
;nevhossz
@local(nev,hossz),
;lokális változók deklarálása
nev:=@inputline("Írd be a neved!"),
;a név bekérése
hossz:=@len(nev),
@prompt("A "&nev&"név"
&@integer(hossz)
&"betűből áll"
```

Az eredmény-string a programot tartalmazó szövegkeretbe íródik. Ez az érték utána szabadon szerkeszthető. Ha azonban a program eredménye numerikus érték, a keret belsejébe nem tudunk belépni.

2. példa. A következő program a billentyűzeten beírt számot négyzetre emeli. Az eredmény most is string, amely a programot tartamazó keret belsejében jelenik meg.

```
;negyzet
@local(x),
x:=@value(@inputline("Írd be a
számot!")),
@prompt(@integer(x)&" négyzete
"&@integer(x*x)
```

Gyakorló feladatok:

1. Írja át az eredeti ALT-F5 makrót úgy, hogy az üzenet magyar nyelvű legyen, s a képernyő alján középre kerüljön!
2. Készítsen programot, amely a billentyűzetről bekért két szám szorzatát kiírja a képernyő alján!

Kóczy A. Judit

Ez a helyes írásvetítő!

Nappali fény mellett is kitűnő képet varázsolnak a vászonra a Polaroid írásvetítők.

A könnyen hordozható legkisebb modell mindössze 4,5 kg.

Előadások látványossá tételéhez ideális útitárs!



Többféle változatban kaphatók a

FLOPPYLAND

számítástechnikai szaküzletben
(Budapest V., Váci utca 84. Telefon/Fax: 118-2651)
és országszerte a Cédrus Rt. viszonteladóinál.

Polaroid

INFORMÁCIÓKÉRÉS: 26 ▲

Mi lett volna, ha...?

Mi lett volna, ha az IBM cég nem találja fel a PC-t? A CP/M és a Zilog Z80 processzor család nőtte volna ki magát 32 bites grafikus környezetté? Esetleg valamelyik jelentős számítógépgyártó cég töltötte volna be a piaci hézagot, létrehozva mondjuk egy NCR személyi számítógépes szabványt? Esetleg maga a Microsoft is csak egy kisebb fejlesztő cég maradt volna, amely időről időre új szoftvertermékekkel jelentkezik?

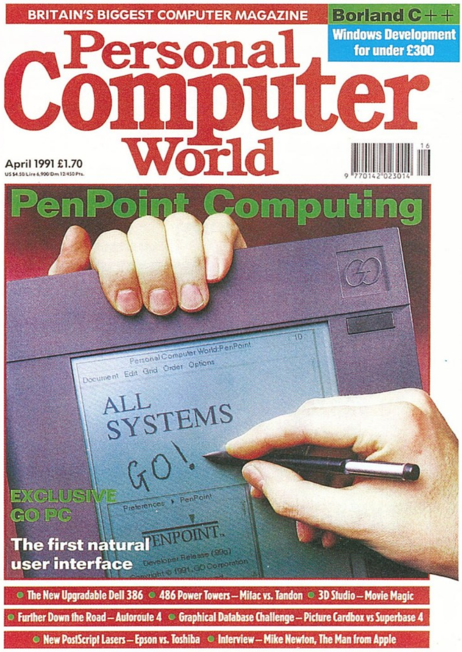
Ragadjuk ki azt az eseményt, amikor az IBM operációs rendszert választott az eredeti PC-hez! IBM-képviselők indultak útnak a floridai Boca Ratonból Menterrey-be, a Digital Research céghez. A CP/M-86 (ez a CP/M 16 bites verziója) licencéről akartak tárgyalni. Mivel nem jutottak megegyezésre a Digital Research vezetőjével, folytatták útjukat a nyugati parton, egészen Seattle-ig. Ide eredetileg a Microsoft Basic szabadalma miatt jöttek, de Bill Gates, a Microsoft alapítója azonnal ráérzett a lehetőségre, és a rá jellemző sebességgel hamarosan meggyőzte az IBM képviselőit arról, hogy új operációs rendszert is ki tud fejleszteni a Basichez. Így került Microsoft környezetbe a 8086-os processzorra alkalmas, CP/M jellegű SCP-DOS operációs rendszer. Ebből az SCP-DOS-ból lett azután a PC-DOS, majd pedig az MS-DOS. Igy alakult ki a PC-s operációs rendszerek területén a Microsoft uralma.

De hogyan alakult volna mindez akkor, ha a Digital Research szerződéskötésre készen várja az IBM küldöttségét? Az biztos, hogy az IBM azonnal aláírta volna a CP/M-86 licenc-szerződését. Ezzel ugyanis mindazokat a szoftverfejlesztőket, akik addig 8 bites CP/M rendszerrel dolgoztak, rávehette volna, hogy térjenek át a PC-re. Nem is igen lett volna más választásuk. Végül is az IBM a CP/M-86-ot alternatív PC-s operációs rendszerként említi. Arra azonban vigyáztak, hogy az árkülönbség akkora legyen a PC-DOS javára, hogy a Digital Research rendszere semmiképpen ne maradhasson versenyben.

Mivel a Microsoft céget az IBM ajánlata előtt az operációs rendszerek nem igazán érdekelték, semmi értelme nem lett volna annak, hogy az IBM megvásárolja tőlük az SCP-DOS-t, ha már a Digital Research szerződése létrejött. Az IBM-nek operációs rendszer témakörben szintén esze ágában nem lett volna olyan céghez fordulni, amely eddig inkább a programozási nyelvekkel foglalkozott, ha egyszer talált olyat, amelynek fő profilja az operációs rendszerek fejlesztése.

Óriási különbség lett volna, ha a CP/M-86 terjed el szabványos PC-s operációs rendszerként. A Digital Research akkor már elkötelezte magát a rendszer multitasking, hálózatos verziója mellett. Nagy valószínűséggel nem kellett volna 1984-ig várnunk arra, hogy megjelenjen egy hálózatban is működő rendszer, s nem születik meg 1987-ben egy olyan multitasking verzió, amely teljesen inkompatibilis. 1984-ben már kompatibilis multitasking rendszer lett volna a teljes PC-piacra vonatkozóan. Ebben az új világban az Apple is átvehette volna az igazi multitasking technológiát és beépíthették volna a Macintoshba is.

Semmi kétség, hogy a PC-s grafikus rendszereket is ugyanarra az időpontra fejlesztették volna ki, csak éppen nem a Windows-t tekintették volna a fejlesztések alapjának, hanem a Digital Research GEM-jét. Az MS-DOS fejlesztés nélkül soha nem lett volna elég pénze a Microsoft cégnek a Windows-projekt kezdeti nehézségeit finanszírozni. Ezzel szemben a Digital Research már 1986-ra kifejlesztett volna a GEM felület multitasking lehetősége mellett valami olyas-



mit is, mint a Windows 3, az új, 32 bites PC-ken is futtatható formában.

A Microsoft cégnek igen sokrétű alkalmazásokat kellett volna létrehoznia ahhoz, hogy egyáltalán a piacon maradhasson. Ma valószínűleg a multitasking GEM-mel használható Microsoft Word és Excel 3 sikerének tanúi lehetnénk, a Windows pedig lemaradt volna. Nem szólva az OS/2-ről, ami csupán egy soha meg nem valósult rémálom lenne!

Természetesen az is lényegesen módosította volna a PC történelmét, ha 1985-ben nem veszti életét egy repülőszerencsétlenség során Don Estridge, az IBM egyik meghatározó személyisége. Ő ugyanis egy éven keresztül úgy vezette a PC-projektet, hogy nem akarta mindenáron kizárni a versenyt, külső technológiát vásárolt és használta a már meglévő személyi számítógépes szabványokat. Ragaszkodott ahhoz az elképzeléséhez, hogy az IBM ne akarja a feltételeket diktálni egy olyan piacon, ahová éppen csak akkor lépett be.

Nehéz elképzelni, hogy Estridge munkássága alatt ugyanúgy elidegenedett volna a PC-világ többi része az IBM cégtől, mint ahogy az történt. Estridge halála után egy régi IBM-es, Bill Lowe lépett helyébe, aki mindent a meglévő IBM-vívmányokhoz akart közelíteni. Ő indította be a PS/2 vonalat, figyelmeztette az esetleges koppintókat a copyright és a szabvány védettségére az MCA adatsín tervével kapcsolatban, forszírozta az OS/2 Extended Editiont, beleértve az IBM szoftvertulajdonjogát a kommunikációra és az adatbázis-hozzáférésre vonatkozóan.

Érdekes abba is belegondolnunk, mennyi olcsó Macintosh-klónt kínálnának ma a piacon, ha valamikor 1989 körül az Apple cég úgy dönt, hogy viszonylag reális áron bárki hozzájuthat a Macintosh Plus ROM-ok licencéhez? A Ma-

cintosh Classic körülbelül ugyanakkorra készült volna el, de a legkülönbözőbb verziókban. Egyes fejlesztésekben gyorsabb processzort vagy nagyobb lemezt találnánk, mások full-page monitorral rendelkeznének, ismét mások jegyzetfüzet formátumúak lennének. Mindegyiken ugyanaz a szoftver futna, és teljes lenne a Macintosh-kompatibilitás. A versenyhelyzet csökkentené az árakat, növelné a teljesítményeket, az alacsony árkategóriába eső Macintoshok szinte minden felhasználói igényt ki tudnának elégíteni. Még a versenytársaktól sem kellett volna tartania az Apple cégnek, ha csak a ROM-licencet adja el, mert számtalan azóta elkészült fejlesztése továbbra is biztosítaná piaci elsőségét. Olyan logikus ez az alternatív verzió, hogy szinte érthetetlen, miért történt másként. Talán azért, mert az Apple továbbra is önmaga akar mindent megtervezni és gyártani, azaz nem akar kiadni a kezéből egy olyan piaci szeletet, amilyent az IBM kiadott a klónok megjelenésével. Akkora azonban a különbség a Macintosh Plus ROM-ok és a Macintosh II család ROM-jai között, hogy bármekkora részesedési veszteség jelentkezne is az alacsony árkategóriában, ezt bőven ellensúlyozná a drágább típusok népszerűségének és ezáltal eladásainak növekedése.

Néhány további ötlet, amellyel mindenki önmaga felépíthet egy kis képzeletbeli történelmi láncot: Mi lett volna, ha az Ashton-Tate időben és tökéletesen működő dBase IV-et mutat be?; ...ha az IBM 1981-ben elárasztja PC-kkel Európát?; ...ha 1985-ben a Microsoft lemond a Windows-fejlesztésekről?; ...ha a Compaq kihoz egy MCA-klónt 1988-ban? Nem biztos, hogy felesleges időtöltés, ha mindezeket végiggondoljuk. Talán közelebb juthatunk az üzleti világ valóságához, ha a fantázia birodalmán keresztül közelítjük meg.

(Personal Computer World, 1991/4)

„Világraszóló" ASCII-utód

Két szabvány verseng azért, hogy a világ összes írásjelét tartalmazó „nemzetközi ASCII" váljon belőlük. Vajon összeütköznek egymással vagy egyezség jön létre köztük?

A világ napról napra kisebb lesz, és egyre nyilvánvalóbb, hogy az ASCII alkalmatlan a világméretűvé váló számítógép-használat kiszolgálására. Az ASCII (American Code for Information Interchange) a hardver és a szoftver közötti mindenféle kommunikáció alapja, mondhatni a számítógépek DNS-e. A 7-bites kódkészlet összesen 128 karakter (betű, írásjel vagy vezérlőkód) meghatározásához elegendő (2^7).

Az ASCII-t 1977-ben fogadta el az Amerikai Szabványügyi Hivatal (ANSI), majd a nemzetközi szerv (ISO) majdnem változatlan formában, ISO 646 jelzéssel szintén beiktatta. Az ISO később 1-bites kiegészítéssel hagyta azt jóvá, ezáltal lehetővé vált további 128 karakteres bővítés (2^8). Ez lett a Latin 1 elnevezésű, 8-bites kódkészlet, amelyben már helyet kaptak néhány európai nyelv (francia, német, spanyol) betűi is, mint például é, ü, ñ. (Érdemes megjegyezni, hogy az ASCII betűszót gyakran használjuk a 8-bites kódkészletre is, ami nem korrekt, mert valójában csak a 7-bites az ASCII.)

Az ISO 646 szabványt afrikai, közel-keleti és távol-keleti országok szintén befogadták és „honosították" saját nemzeti karakterkészletükre, vagy létrehoztak hasonló szabványokat. Eközben az egyes országokon belül is számos eltérő gyártói kódkészletet alakítottak ki. Azoknak a cégeknek, amelyek nemzetközi méretekben működtek, éppen ezért vált lidércnyomássá a kóddzsungelben való eligazodás. Valamit kellett tenni.

1983-ban az ISO elkezdte egy új szabvány kidolgozását a karakterek kódolására. Az ISO 10646 szabvány elkészítésének eredeti célja egy 2-bájtos (16-bites) készlet létrehozása volt, benne 65 536 karakterrel — ami bizonyosan több, mint amennyi kódra valaha is szükség lesz. Az ANSI egyik szakbizottsága most azon dolgozik, hogy az új 16-bites szabvány kompatibilis legyen az összes jelenlegi nemzetközi szabvánnyal. Az első 128 karakter megegyezik majd a régi jó ASCII-val, utána pedig jön a többi karakterkészlet.

128-ról 4 milliárdra

Két probléma azonban gyorsan felszínre került. Az egyik, hogy a már meglévő nemzetközi szabványok mindegyike tartalmaz vezérlőkódokat is, amelyek például a nyomtatónak „üzennek" (sor vége, lapdobás stb.). Sok érv szólt amellett, hogy megtartsák ezeket a kódokat. A kommunikációs berendezések ugyanis a vezérlőkarakterek alapján döntik el, hogy mit kezdjenek a kapott adatokkal. Ezek megváltoztatása az új kódkészletet inkompatibilissá tenné a világszerte használt régi kódkészletekkel. A probléma a következő: ha fenntartjuk a helyet mindazoknak a karaktereknek, amelyek vezérlőkódként számításba jöhetnek, akkor a rendelkezésünkre álló 65 536 helynek máris felhasználnánk mintegy 40 százalékát.

A másik probléma a kínai, japán és koreai írással függ össze. Ezek tartalmazzák ugyanis a legtöbb jelkombinációt a világ összes írása közül, mert szavakat, jelentéstartalmakat is kifejeznek velük (képírási jelek). E három írás karaktereinek egy része nemcsak nagyon hasonlít egymásra, hanem jelentésükben is megegyeznek. Helytakarékosságból a nemzetközi szabványon dolgozó bizottság javasolta az ismétlő-



dések kiiktatását, alapnak tekintve a közös jeleket tartalmazó egységesített kínai (han) jelkészletet és külön kezelve a nemzeti kiegészítéseket. Ezt azonban a japánok és a koreaiak nem fogadták el. Ha viszont a kódkészletben benne marad a sokezernyi ismétlődés, akkor a 2 bájt egyszeriben nem elegendő minden kód tárolására.

A bizottság több mint egy éven át birkózott a kódkészlet méretének növelésével. Elég lesz 3 bájt? Túl sok lenne a 4? Végül is a 4 bájt (32 bit) mellett döntöttek — amire bőven elfér több mint 4 milliárd karakter is (2^{32}). A számítástechnikusok egy része tiltakozott, hogy minden adatkommunikációban négyszer annyi ideig tartana egy közönséges szövegállomány továbbküldése. No nem egészen — mondták —, mert az első 3 bájt ugyanannak az ábécének a kódkészletében teljesen megegyezne, azokat nem kellene karakterenként mindig továbbítani. Túl bonyolult — mondták erre azok, akik inkább az állandóan azonos hosszúságú kódokat részesítenék előnyben.

Unicode

1987-ben három amerikai kutató, Joe Becker, Lee Collins és Mark Davies elkezdett kidolgozni egy kódkészletet, amely egyszerűbb és egyöntetűbb az ISO 10646 szabványnál, mégis tartalmazza a nemzeti ábécékhez szükséges összes karaktert. Korábban Becker és Collins együtt dolgozott a legendás Xerox Star számítógépen (a Macintosh előfutárán), s annak számos innovatív megoldása között volt egy 16-bites többnyelvű kódszabvány is. Utólag kiderült, hogy ez nagyon jó kezdeményezés volt, csak akkor az Apple a karakterkódok helyett inkább a betűtípusok iránt kezdett érdeklődni, a Xerox pedig nem folytatta a saját gépek és operációs rendszerek fejlesztését — ehelyett mindketten átvették a szakmában általánosan elfogadott kódszabványokat. Csak utólag jött létre egy kis csapat, hogy a korábbi jó tapasztalatok felhasználásával létrehozza a kódok új generációját.

Az új szabvány neve Unicode lett. Sok helyen bemutatták az ISO szabvány 16-bites vetélytársát, és a tervezeten dolgozó gárda 1991 januárjában vállalattá is alakult, tekintélyes cégek támogatásával (Metaphor, Microsoft, Sun, Adobe, Claris, Next, Aldus, Go, Lotus, Novell stb.). A kód főbb követelményeit és céljait az alábbiakban foglalták össze:

— **Teljesség.** Mindenféle szöveg karaktereinek teljes tárháza legyen, beleértve olyan nyelvekét is, amelyek karakterei nincsenek maradéktalanul rögzítve, vagy a holt nyelvekét, mint például a szanszkrit.

— **Egyszerűség és hatékonyság.** Minden Unicode karakterkód ugyanolyan hosszú (16 bit) és meghatározott karaktert jelent, nincsenek benne vezérlőkarakterek, kódbővítők, megszakítók, amelyek megzavarhatják a számítógépet és a karaktersorozatok értelmezését.

— **Félreérthetetlenség.** Minden kód félreérthetetlenül egyetlen karaktert jelent, s annak olvasási problémája nem gördíthető tovább úgy, hogy az utána következő karakterek jelentése torzuljon.

— **Korrektség.** A szabványba felvett minden karakter helyességét nyelvi szakemberekkel kell ellenőriztetni.

— **Hitelesség.** A szövegadatokból semmi nem veszhet el és nem módosulhat, amikor a meglévő kódkészletek és az Unicode közötti konverzióra kerül sor.

Csak ne mind a kettőt!

Most tehát gyakorlatilag két kódkészlet várja, hogy elfogadják nemzetközi szabványként. Tökéletes terv természetesen nincs, és egyidejűleg minden követelményt nem is lehet teljesíteni. Például annak érdekében, hogy az eredeti ASCII kódtábla kompatibilitását megőrizzék, az Unicode is fenntartja a helyet az eredeti ASCII szabványban lévő 65 vezérlőkódnak, bár azokat az Unicode nem használja, és továbbiakat nem is ismer el. Ezzel szemben az ISO 10646 valamennyi előző kódtábla vezérlőkaraktereinek helyét fenntartja, összesen 28 672-t.

Az Unicode szerkezete

Kódok	Karakterek
65 025 — 65 536	Kompatibilitási terület
59 392 — 65 024	Felhasználói terület
16 384 — 59 391	Kínai, japán, koreai képírásjelek (Egységesített han karakterek) Későbbi felhasználásra fenntartva
12 288 — 16 383	Kínai, japán, koreai segédírásjelek (Ábécé és interpunkció)
8 192 — 12 287	Jelek (Írásjelek, matematikai műveletjelek, szimbólumok, díszítő jelek)
0 — 8 191	Ábécék (Angol, Latin1, európai, kiterjesztett Latin, fonetikai, görög, cirill, örmény, héber, arab, etiópiai stb.)

Az Unicode különbséget tesz egy karakter kódja és díszítése (konkrét megjelenítése) között. Ez utóbbi betűtípusonként változik, de a betű lényeges vonásai mindig azonosak maradnak. (A mellékelt tábla mutatja az Unicode felépítését.) Az ISO kódkészletben minden ékezetes betűnek van külön kódja. Az Unicode is tartalmazza ezeket a kódokat, de ad egy kódot külön az ékezeteknek is, amelyek az ékezet nélküli betűkkel kombinálhatók.

Nemzetközi kódkészletet alkotni nem abból áll, hogy azonosítjuk a karaktereket és beszámozzuk őket. Még az ASCII esetében is — amely igazán stabil és mindenki számára érthető rendszer — hiányzik a koncepció a sorok végének és a bekezdések végének egységes jelölésére, s már ebből is rengeteg adatátviteli zavar származik.

Milyen hatással lesz a számítástechnikai iparra a „multibájtos" szabványok bevezetése? Mindkét kódkészlet támogatói azt várják, hogy a rendszerek, szoftverek nemzetközi értékesítése könnyebb lesz. A fejlesztőknek viszont olyan termékeket kell előállítaniuk, amelyek változtatás nélkül szállíthatók akár Európa bármelyik országába, akár Japánba vagy Kínába. A hardvert illetően, nem követel-e az új kódkészlet több RAM-ot és merevlemez-kapacitást? De igen, bár nem olyan nagyon sokat, mint gondolhatnánk. A RAM-ban a karakterek tárolására csak plusz 10-20 százalékot használnának fel. Igaz viszont, hogy a tiszta szövegállományok mérete megduplázódna.

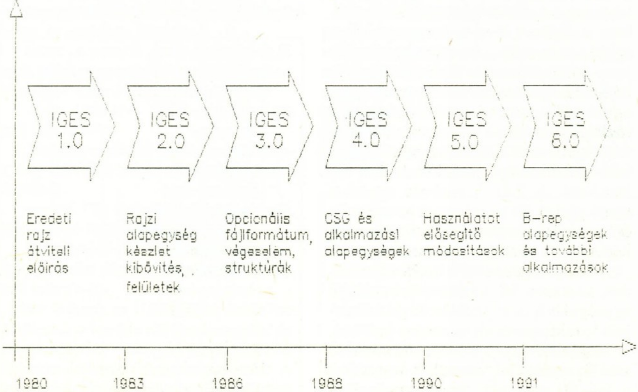
A két kódszabványt valahogy közös nevezőre kellene hozni, mert egyelőre az amerikai ipar általában az Unicode-ot támogatja, a többi ország szakértői pedig az ISO 10646 szabvány mellett foglalnak állást. Ha nem sikerülne valamelyiket — a szükséges kompromisszumok árán — egységesen elfogadtatni, ha tehát a felhasználónak mindkettőt használnia kellene, az lenne a legrosszabb megoldás. De ezt szerencsére senki nem akarja.

(Byte, 1991/7)

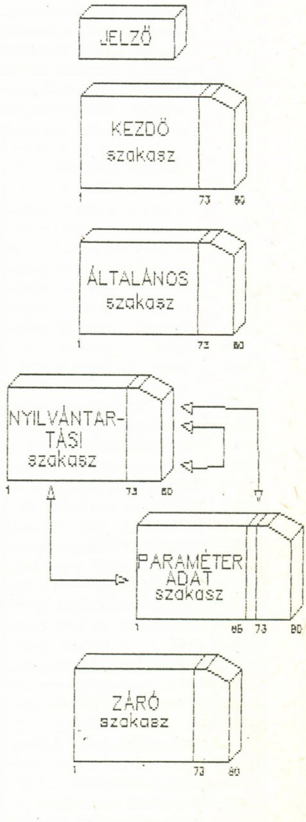
Ami a gyakorlatban is bevált

Az Egyesült Államokban 1980-ban dolgozták ki és 1981-ben szabványosították az IGES adatleírási és -átviteli szabvány első változatát (ANSI Y14.26M). Az IGES lényegében szabványosított adatállomány-formátum, amelynek legfontosabb sajátossága a kötött funkcionális alapegységkészlet, a fájlnak a szakaszokból adódó belső strukturáltsága és felhasználási metodikája. Eredetileg rajzi adatok átviteli eszközének tekintették, később azonban e koncepció jelentősen módosult — bár a neve (Initial Graphics Exchange Specification) még ma is ezt tükrözi. Újabb változatai geometriai és nem geometriai adatok CAD/CA... rendszerek közötti cseréjét egyaránt lehetővé teszik kettős adatkonverzió alkalmazásával.

Az IGES kommunikációs sémája és szemléltetési struktúrája számos előnyt nyújt, ugyanakkor néhány kedvezőtlen (mondhatni hátrányos) sajátossággal még mindig számolni kell. Végső soron ez az oka, hogy az IGES-t folyamatosan karbantartják és tökéletesítik. Létezésének tíz éve alatt hat változata született (1. ábra), amelyek egymással felülről kompatibilisak. A fogyatékosságok — főleg a korai változatokban — könnyen felismerhetők, ha az újabb koncepciók szerint fejlesztett szabványokkal, például a PDES-szel vagy a STEP-pel vetjük össze. Mindamellett még e későbbi szabványoknak is vannak olyan gyenge pontjai, amelyek tekintetében az IGES előnyösebb.



1. ábra. Az IGES fejlődésének állomásai



2. ábra. Az IGES fájl szakaszai

Az IGES fizikai fájl

Az IGES fájl tartalma a felhasználástól függetlenül ASCII formátumban szerkeszthető, ami széles körű kompatibilitást eredményez. Az IGES 5.0 specifikációja szerint három fájlformátum lehetséges:

1. Közönséges ASCII szöveg.
2. Tömörített ASCII szöveg.
3. Bináris (bitfüzér).

Az IGES egyik jellemző, ma már örökletes fogyatékosságának lehet te-

kinteni a 80 karakterhelyes IBM lyukkártya-formátumból eredő terjengősséget. Ez elsődlegesen abból adódik, hogy az üres helyeket az ASCII kódtáblázat szerinti 0010000 üreshely karakterrel kell feltölteni. Az IGES minden számot karakter ábrázolásúvá alakít át. Ez a konvertálásnál többletidőt igényel, ugyanakkor előnyként jelentkezik a szövegfájl egyszerű feldolgozhatósága.

Az IGES fizikai fájl egy legelől opcionálisan megadható és öt kötelező sorrendű és tartalmú szakaszból áll (2. ábra). A szakaszokat a kötött helyen elhelyezendő szakaszmutató jelöli. Az opcionális JELZŐ szakaszt akkor lehet megadni, ha bináris vagy tömörített fájlformátumot írunk elő. Az első kötelező szakasz a KEZDŐ, amely a feladatra vonatkozó általános információkat tartalmazza. A második szakasz az ÁLTALÁNOS, ami az alkalmazott előfeldolgozóra vonatkozó és az utófeldolgozó által igényelt információkat tartalmazza. A NYILVÁNTARTÁSI szakasz a fájlban lévő valamennyi alapegység azonosító információját megadja — egyenként 2 x 80 karakteren. A PARAMÉTERADAT szakasz az alapegységekhez tartozó paraméteradatokat foglalja magában. A paraméteradatok formátuma az IGES által rendelkezésre bocsátott formátumokból szabadon választható. A ZÁRÓ szakasz egyetlen sort foglal el. Ez a szakaszok azonosítóit és az egyes szakaszokban felhasznált legnagyobb sorszámokat tartalmazza.

A közönséges ASCII IGES fájl minden szakasza 80 karakter hosszúságú egységekből (sorokból) épül fel. A sorbeli karakterpozíciót az oszlop jelöli ki. A megfelelő szakaszazonosító minden sor 73. oszlopába bekerül. A sorszámot a 74—80. oszlopok tartalmazzák. Egy rekord egy vagy több sorból állhat. A harmadik és a negyedik szakaszokban az alapegységekhez tartozó paraméterek sorozatát foglalja magában. A tömörített formátumú fájl nyilvántartási és paraméteradat-szakaszainak szervezésére és adatokkal való feltöltésére az IGES specifikáció egyéb előírásokat ad meg.

A közönséges ASCII formátumú fájl kezdő szakaszában legalább egy rekor-

1. táblázat

A geometriai alapegységek választéka

* Körív
* Összetett görbe
* Kúpszelet
* Összetartozó adatok
- Középvonal
- Egyszerű zárt terület
- Lineáris pálya
- Nézési vonal
- Metszésvonal
* Sík
* Vonal
* Parametrikus szplájn görbe
* Parametrikus szplájn felület
* Pont
* Vezérgörbés felület
* Forgásfelület
* Tabulált henger
* Transzformációs mátrix
* Jelzőfolt
* Racionális B-szplájn görbe
* Racionális B-szplájn felület
* Párhuzamos görbe
* Csatoló pont
* Csomópont
* Végeselem
* Csomópont-elmozdulás és
* Elfordulás
* Párhuzamos felület
* Görbe
* Parametrikus felület
* Csonkolt parametrikus felület
* Csomóponti eredmények
* Elemeredmények
* Hasáb
* Egyenes prizma
* Egyenes körhenger
* Csonkolt egyenes körkúp
* Gömb
* Tórusz
* Forgástest
* Lineárisan pásztázott test
* Ellipszoid
* Bináris fa
* Testösszeállítás
* Testpéldány

```
IGES file generated from an AutoCAD drawing by the IGES                 S0000001
translator from Autodesk, Inc., translator version IGESOUT-2.0.         S0000002
,,6Hselejt,10HSELEJT.IGS,10HAutoCAD-10,11HIGESOUT-2.0,16,38,6,99,15,6HseG0000001
lejt,1.0,1,4HINCH,32767,3.2767D1,13H910505.041218,1.0D-8,420.0,6HThroop,G0000002
14HAutodesk, Inc.,4,0;                                                  G0000003
     110       1       1       1                               00000000D0000001
     110                       1                                       D0000002
     100       2       1       1                       0       00000000D0000003
     100                       2                                       D0000004
     124       4       1                                       00010000D0000005
     124                       1       0                               D0000006
     110       5       1       1                               00010000D0000007
     110                       1                                       D0000008
     110       6       1       1                               00010000D0000009
     110                       1                                       D0000010
     110       7       1       1                               00010000D0000011
     110                       1                                       D0000012
     110       8       1       1                               00010000D0000013
     110                       1                                       D0000014
     102       9       1       1                       0       00000001D0000015
     102                       1                                       D0000016
110,90.0,230.0,0.0,220.0,260.0,0.0;                                    1P0000001
100,0.0,260.0,210.0,2.882842712474602,210.0,2.882842712474602,         3P0000002
210.0;                                                                 3P0000003
124,1.0,0.0,0.0,0.0,0.0,-1.0,0.0,0.0,0.0,0.0,-1.0,0.0;                 5P0000004
110,180.0,1.0D2,0.0,1.6D2,200.0,0.0;                                   7P0000005
110,1.6D2,200.0,0.0,6.0D1,1.8D2,0.0;                                   9P0000006
110,6.0D1,1.8D2,0.0,8.0D1,8.0D1,0.0;                                  11P0000007
110,8.0D1,8.0D1,0.0,180.0,1.0D2,0.0;                                  13P0000008
102,4,7,9,11,13;                                                      15P0000009
S0000002G0000003D0000016P0000009                                        T0000001
```

3. ábra. Példa IGES fájlra

2. táblázat

Az annotációs alapegységek

* Összetett adatok
— Középvonal
— Metszet
— Nézetvonal
* Szögméret
* Átmérőméret
* Keretes felirat
* Általános címke
* Általános felirat
* Mutatóvonal (nyíl)
* Hosszméret
* Ordinátaméret
* Pontméret
* Sugárméret
* Általános szimbólum
* Metszett terület

dot el kell helyezni. Formátumra vonatkozó megkötés nincs, viszont az ASCII karakterkészlet használandó. Az általános szakaszban csak az előfeldolgozóra és az állomány felhasználására vonatkozó információk helyezhetők el. Az összesen 24 globális paraméterből az első kettő a paraméterhatárolót és a rekordhatárolót adja meg, ha azok az alapértelmezéstől (vessző és pontosvessző) eltérnek. A nyilvántartási szakasz minden alapegységre húsz nyolckarakteres mezőt tartalmaz. A paraméteradat szakaszban az első mezőt feltöltő alapegységkód után az adatok egészen az üresen maradó 65. oszlopig tetszőleges formátumban beírhatók. A 66—72. oszlopok az alapegység nyilvántartási szakaszbeli első sorának számát tartalmazzák. A rekordhatároló után bármilyen megjegyzés elhelyezhető. A záró szakasz tíz nyolckarakteres mezőt foglal magában, amiből az első négy és a tizedik használt. A tömörített formátum esetén a kezdő, az általános és a záró szakaszok változatlanok, viszont a nyilvántartási és a paraméteradat-szakaszok egyetlen ADAT szakaszba olvadnak össze. Külön előírás vonatkozik a bináris formátumú fájlok feltöltésére és értelmezésére.

Ami körül a világ forog

Az IGES 5.0 változata a következő funkcionális alapegységeket kezeli:

* geometriai alapegység (az egyszerű vagy összetett geometriai elemeket írja le);
* annotációs (a rajzi segédeszközöket írja le);
* tulajdonság (valamely alapegységhez numerikus vagy szöveges információt kapcsol);
* kapcsolati (több alapegységet logikailag egymáshoz kapcsol);
* nézet (a termék geometriai modelljének rajzi vagy képi szemléltetéséhez szükséges információkat hordozza);
* rajzi (nézetek rajzi elrendezését teszi lehetővé);
* transzformációs mátrix (az alapegység eltolását vagy elforgatását teszi lehetővé);
* makró (új összetett alapegységek definiálását teszi lehetővé);
* felhasználó által definiált alapegység (speciális alapegységek kijelölését teszi lehetővé).

Mint látható, az IGES fájl a modellre vonatkozó információkat geometriai és nemgeometriai jellegű alapegységek rendezett halmazaként tartalmazza. A geometriai alapegységek a fizikai alak elemei, például pontok, vonalak, felületek stb. Az IGES a legtöbb CAD/CAM rendszer által kezelt geometriai alapegységek formális leírását megadja. A geometriai alapegységek választékát az 1. táblázat adja meg. Az IGES első változatai nem kezelték az összes felület- és testmodelltípust. Az 1983-ban kibocsátott második változat az elsővel összevetve jelentős geometriai alapegység-kezelési kiterjesztéseket foglal magában. Az 1986-ban megjelent harmadik változat kiegészült végeselemes modelladatok kezelésének lehetőségével. Megoldották a kúpszeletek, szplájnfelületek, szögméretek stb. kezelésének problémáját. Kiegészítették többszörös transzformációs alapegység-kezelési lehetőséggel is, emellett a testkezelés további finomítását is végrehajtották.

3. táblázat
A strukturális alapegységek
* Null
* Kapcsolat definiálása
* Vonaltípus definiálása
* Makródefiniálás
* Alábra definiálása
* Szövegtípus definiálása
* Szövegmegjelenítési sablon
* Színdefiniálás
* Hálózati alábra definiálása
* Jellemzőtáblázat definiálása
* Kapcsolati példány
* Rajz
* Jellemző
* Egyedi alábrapéldány
* Nézet
* Négyzetes tömb alábra példánya
* Körkörös tömb alábra példánya
* Külső hivatkozás
* Csomóponti terhelés/korlátozás
* Hálózati alábra példánya
* Jellemzőtáblázat példánya

Az 1988 végén gyakorlati hasznosításra bevezetett negyedik változat további lehetőségeket nyújt a szabadformájú felületleírás terén. Az 1990-ben megjelent ötödik változat a geometriai alapegységek tekintetében a CSG modelladatok és struktúrák leírási sémáival bővült. Vagyis e változat összességében 2D rajz-, 3D huzalvázmodell, 3D szplájnfelület, CSG testmodell, valamint végeselemes modell leképezésére és átvitelére alkalmas. A B-rep adatok kezelését és a kapcsolódó topológiai struktúrák kezelését csak a 6.0 változatban tervezik.

A fájlban az alapegység-nyilvántartási számok 0001 és 0599, illetve 0700 és 5000 között lehetnek. Az alapegység típusazonosítói 0600-tól 0699-es értéket vehetnek fel. Ugyanakkor a felhasználó által definiált egységekre a 10000-től 99999-ig terjedő azonosítók alkalmazhatók. A geometriai alapegységek kódjai rendre 100-tól 184-ig terjednek, létezik a 125-ös kód, és ide tartozik a 430-as kód is, de kimarad a 166, a 170—178 és a 182 kód.

A testmodellezéshez felhasználható alapegységek a geometria- és a struktúrameghatározó alcsoportokba oszthatók. A kilenc geometriai alaptest mint alapegység a szemantikai tartalmat, a méreteket, a helyet és a hozzárendelt transzformációs mátrixot fejezi ki. A bináris fa alapegység a fa elemeire és a CSG műveletekre (egyesítés, kivonás és közös rész) ad meg információt. A fa elemei lehetnek alapegységek, más bináris fák és CSG testpéldányok. A testösszeállítás meghatározott geometriai kapcsolattal rendelkező CSG testek összessége. A testmodell leírása aciklikus irányított gráffal történik.

Az IGES a nem geometriai alapegységeknek két típusát, az annotációs alapegységeket és a struktúrális alapegységeket definiálja (2. táblázat). A megfelelő alapegységkódok 200-tól 499-ig terjednek. Ide tartozik még a 106-os kód az összetartozó adat alapegység kódja is. A nem geometriai alapegységek attribútumleíró, annotációs, méretmegadási stb. funkciókat töltenek be. Az annotációs alapegységek vagy egyediek, vagy más alapegységekből szerkesztettek. A definíciós térben vagy a rajzi alapegységhez kapcsolt 2D térben adhatók meg.

A kapcsolati struktúrát (sémát) az előfordító a kapcsolatdefiniálási alapegységen keresztül jelöli ki. A definíciós séma lehetővé teszi, hogy osztá-

4. táblázat
Az előre definiált kapcsolati formák
* Csoport
* Külső logikai hivatkozású fájl indexe
* Látható nézetek
* Látható nézetek, szín- és vonalvastagság
* Alapegységcímke megjelenítése
* Csoport-visszamutató pointer nélkül
* Egyszeres szülői kapcsolat
* Külső hivatkozású fájlindex
* Méretezett geometriai kapcsolat
* Rendezett csoport-visszamutató pointerrel
* Rendezett csoport-visszamutató pointer nélkül
* Síkbeli kapcsolat
* Áramlás

lyoknak nevezett többszörös adatcsoportokat hozzunk létre. Minden osztály esetében a séma mondja meg, hogy visszafelé mutató pointerekre szükség van-e. E pointerrel hivatkozik a kapcsolati példány definiálására a kapcsolatban szereplő tagalapegység. Minden kapcsolati példány eltérő számú alapegységet foglalhat magában. A fájlban szerepelhet előre definiált, de meg nem jelenő kapcsolat (alapegységkód 1—5000) vagy az előfordító által definiált kapcsolat (alapegységkód 5001—9999). Ez utóbbit egyszer kell definiálni, de többször felhasználható. A definiálás kiterjed a kapcsolati formára, az osztálydefiníciók számára, az alapegységek típusára, számára és az esetleges visszamutató pointerekre. A rendelkezésre álló strukturális alapegységeket a 3. táblázat mutatja. Viszont az előre definiált kapcsolatok megengedett formáit az IGES specifikáció a 4. táblázat szerint rögzíti.

A makródefiniálási alapegység paraméteradat szakasza tartalmazza a makró törzsét. A makrótörzsben tizenegy nyelvi utasítás alkalmazható. A makró akkor használható, ha a példánya megtalálható a fájlban. A makródefiníció nyilvántartási bejegyzés rekordja szabványos formátumú. A makrónyelv elemei konstansok, változók, egész szám, valós, duplapontosságú és string függvények. Feltételes kifejezések is képezhetők. Hozzá kell tenni, hogy a makróképzést az IGES 5.0 változata a korábbi használati gyakoriság alapján szükségtelennek nyilvánítja.

Ami érdekes az IGES koncepcióban, hogy a termék jellemzőinek megadását egyetlen alapegység (a 406-os) különböző formaszámaihoz kapcsolja. A kezelhető jellemzőket az 5. táblázat foglalja össze. Az anyagjellemző paramétereket a 406-os alapegység 11-es formaszáma fedi le. Ezeket a 6. táblázat mutatja.

5. táblázat

A termékjellemzők leírására használható információk

* A definíciós szintek
* A tartomány korlátozása
* A szintekre vonatkozó funkció
* A vonalvastagítás
* Fúrt lyuk
* A hivatkozáskijelölő
* Az érintkezőszám
* Az alkatrészszám
* A hierarchia
* Táblázatos anyagra jellemző adatok
* Külső hivatkozású fájlok listája
* Névleges méret
* Az áramlási út specifikációja
* Név
* Rajzméret
* Rajzi mértékegységek
* Karakterek közötti térköz

Szólni kell még az IGES végeselemes modellek adatainak leírását és átvitelét támogató szolgáltatásairól. A modell-leírást a különböző végeselemek könyvtárát magában foglaló alapegység, a csomópontokat meghatározó és megjelölő alapegység, a csomóponti terheléseket megadó alapegység, valamint a csomóponti elmozdulásokat leíró alapegység segítségével valósítja meg. Az elemkönyvtár alapegység gyakorlatilag mindent lefed, még a speciális elemtopológiákat is.

Jó, ha emlékszünk rá

Minden alapegység előfordulásához pontosan egy nyilvántartási bejegyzés és egy paraméteradat bejegyzése tartozhat. A nyilvántartási bejegyzés kötött mezőket foglal el, a paraméteradatok alapegységfüggőek, vagyis formátumuk és hosszuk változik. Az alapegység-nyilvántartás és a paraméteradatok összerendelése kétirányú pointerezéssel történik. Minden alapegységhez egyedi alapegységkódszám tartozik. A típuson belüli megjelenési formákat formaszám különbözteti meg. A felületek kivételével az alapegységek függetlenek egymástól.

Alapegységek gyűjteményének rendezett kezelésére az egymásba ágyazható alábrák állnak rendelkezésre. Az objektumok közötti logikai és a lokációból adódó fizikai viszony formátumozott megadására a kapcsolat használható, amely előírható a csatlakozó pontok helyével, az áramlási úttal és a fizikai kapcsolattal. A kapcsolatok tényleges leírását a megfelelő alapegységek alkalmazása teszi lehetővé.

Az IGES megkülönbözteti a modellteret és a definíciós teret. A modelltér háromdimenziós, jobbsodrású derékszögű koordináta-rendszer, amely a modellhez kötött. A definíciós tér ugyancsak jobbsodrású derékszögű koordináta-rendszer, mely modellről modellre változik. A modelltér koordinátái a definíciós tér koordinátáiból egy forgatási mátrix és egy eltolási vektor segítségével állíthatók elő. A modelltérben minden alapegység irányított.

Az IGES lehetővé teszi, hogy a felhasználó a létező alapegységekből új alapegységet hozzon létre. Ha nagyon szükséges, makróalapegység is definiálható, de akkor minden előfordulásához egy példányt is meg kell adni. Az aktuális IGES munkafájlhoz tartozó fájlok nevének megadásával, illetve a külső hivatkozást biztosító alapegységgel lehetőség van a definíció szempontjából idegen alapegységek felhasználására is. Minden hátránya ellenére lehetőleg maradjunk a teljes ASCII formátumnál, mert a Magyarországon létező rendszerek szinte kizárólagosan ezt használják.

6. táblázat

Az anyagjellemzők leírására használható információk

* Rugalmassági modulus
* Poisson-féle tényező
* Nyírási modulus
* Anyagjellemző mátrix
* Fajlagos sűrűség
* Hőtágulási együttható
* Rétegezett anyag merevségi mátrixa
* Hajlított anyag merevségi mátrixa
* Keresztirányban nyírt anyag merevségi mátrixa
* Összetett igénybevételű anyag merevségi mátrixa
* Anyagi koordináta-rendszer
* Csomóponti terhelésre/megtámasztásra vonatkozó adatok
* Gerendaelem keresztmetszeti jellemzője
* Gerendaelem véglapjának szabadságfokai
* Szélsőszál-távolságok
* Feszültségszámítási helyek
* Elem vastagsága
* Nem szerkezeti tömeg
* Hővezetési együttható
* Hőkapacitási tényező
* Vezetőfilm-együttható
* Elektromágneses sugárzási paraméterek

A fejlett ipari országokban tapasztalható széles körű elfogadás miatt célszerű hazánkban is az IGES specifikációt alkalmazni különböző CAXX alkalmazói szoftverek közötti adatkommunikációban.

Bár az elő- és utófeldolgozók programozása nem kis időt, hozzáértést és találékonyságot igénylő tevékedés, a ráfordítások az integrálás lehetőségének megteremtésén keresztül minden szempontból megtérülnek. És ez igaz akkor is, ha az IGES-szel több dolog egyértelműen nem oldható meg. Jelenleg nincs jobb. De várhatólag lesz. Mert a STEP már óvatosan lépked előre. Azt, hogy milyen LÉPÉS is a STEP, a következő számban foglalom össze.

Horváth Imre

A GEM operációs rendszer XIII.

Ígéretünkhöz híven sorozatunkat a GEM VDI (Virtual Device Interface) függvényeinek ismertetésével folytatjuk.

Mielőtt azonban rátérnénk a részletes ismertetésre, nézzük meg, hogy ez a programrészlet hol helyezkedik el a rendszerprogram részegységeinek a hierarchiájában.

A TOS rendszerprogram két fő részből áll:

1. DOS (BIOS, XBIOS, GEMDOS),
2. GEM (AES, VDI).

A VDI olyan grafikus I/O rendszerekhez szabványosított általános interfész, amely a programozás során hardverfüggetlenséget biztosít. A vektor- és a pixelgrafika alkalmazását egyaránt támogatja, gyakorlatilag egy óriási grafikai funkciógyűjtemény.

A VDI maga is két részből áll, a GDOS-ból (Graphics Device Operating System), azaz a grafikus eszközök működtető rendszeréből és a GIOS-ból (Graphic Input Output System), azaz a grafikus I/O rendszerből). A GDOS a VDI hardverfüggetlen része, amelynek a következő fontosabb funkciói vannak:

— Meghajtóprogramok kezelése.
— Karakterkészletek (fontok) kezelése.
— Az RC és az NDC koordinátarendszerek közötti transzformálás (lásd sorozatunk XI. részét).

A GIOS az eszközspecifikus meghajtóprogramok összessége. Feladata a GDOS és a konkrét fizikai periféria közti kapcsolat biztosítása.

(A hierarchikus felépítést a mellékelt ábra szemlélteti.)

A VDI-függvények a következő hét nagy funkciócsoportba sorolhatók be:

1. Kontrollfunkciók. Ezek a függvények végzik el a grafikus munkaállomás kezdőértékeinek beállítását (inicializálás).
2. Outputfunkciók. E függvények által kapjuk meg a grafikus formátumot.
3. Attribútumfunkciók. Segítségükkel állíthatjuk be a kiszínező minta, a vonal, a szöveg stb. attribútumait.
4. Raszterfunkciók. Felhasználásukkal kezelhetjük a pixelgrafikus blokkokat.
5. Inputfunkciók. E függvényeken keresztül kommunikál a rendszer a felhasználói programokkal.
6. Lekérdező funkciók. Segítségükkel a beállított paraméterek és attribútumok értékeit kaphatjuk meg.
7. Escape-ek. Speciális funkciók. Például a szöveges képernyőn a kurzort pozicionálhatjuk velük.

Változónevek, paraméterátadások

Ötfajta típust használnak a függvényhívások C programnyelven:

— Control (contrl[ ]).
— Input paraméterek (intin[ ], integer input).
— Input pontkoordináta (ptsin[ ], points input).
— Output paraméterek (intout[ ], integer output).
— Output pontkoordináták (ptsout[ ], points output).

Input paraméterek:
contrl[0]: a funkció operációs kódja.
contrl[1]: az input koordinátapárok száma.
contrl[3]: az intin változók száma (mezőhossz).
contrl[5]: alkód.
intin[ ]: az input mező integer típusban.
ptsin[ ]: az input mező pontkoordinátákkal.

Output paraméterek:
contrl[2]: az output koordinátapárok száma.
contrl[4]: az intout változók száma (mezőhossz).
intout[ ]: az output mező integer típusban.
ptsout[ ]: az output mező pontkoordinátákban.

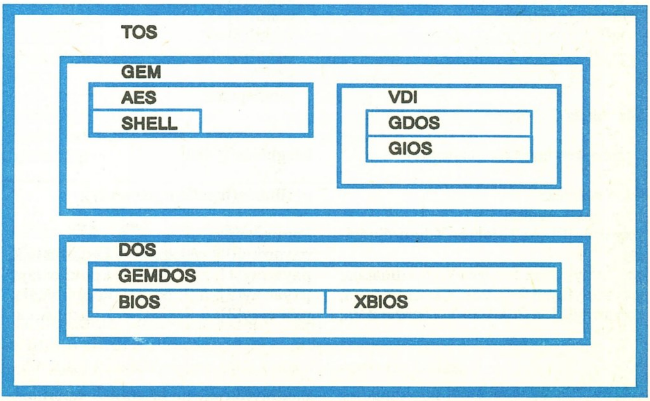
Egyéb I/O paraméterek:
contrl[6]: a munkaállomás számára foglalva.
contrl[7-n]: kódspecifikus kiegészítő információk.

Függvények

A sort az output függvényekkel kezdjük.

POLYLINE (VDI 6)

Egy vonallal sorban összeköti a megadott pontokat.



Az operációs rendszer részeinek hierarchikus felépítése

Deklaráció C-ben:

```
void v_pline(handle,n,pxyarray)
  int handle,n,pxyarray[ ];
{
  pioff=pxyarray;
  contrl[0]=6;
  contrl[1]=n;
  contrl[3]=0;
  contrl[6]=handle;
  vdi( );
  pioff=ptsin;
}
```

Meghívás C-ből:

```
v_pline)handle,n,pxyarray);
```

Paraméterek:
pxyarray[0]: az 1. pont X koordinátája;
pxyarray[1]: az 1. pont Y koordinátája;
pxyarray[2]: a 2. pont X koordinátája;
pxyarray[3]: a 2. pont Y koordinátája;
.
.
.
pxarray[2n-2]: az n. pont X koordinátája;
pxarray[2n-1]: az n. pont Y koordinátája.

POLYMARKER (VDI 7)

Jelzéseket, markereket rajzol a megadott pontokba.

Deklaráció C-ben:

```
void v_pmarker(handle,n,pxyarray)
  int handle,n,pxyarray[ ];
{
  pioff=pxyarray;
  contrl[0]=7;
  contrl[1]=n;
  contrl[3]=0;
  contrl[6]=handle;
  vdi( );
  pioff=ptsin;
}
```

Meghívás C-ből:

```
v_pmarker(handle,n,pxyarray);
```

Paraméterek:
n: jelzések száma;
pxyarray[0]: az 1. jelzés X koordinátája;
pxyarray[1]: az 1. jelzés Y koordinátája;
pxyarray[2]: a 2. jelzés X koordinátája;
pxyarray[3]: a 2. jelzés Y koordinátája;
.
.
.
pxarray[2n-2]: n. jelzés X koordinátája;
pxarray[2n-1]: n. jelzés Y koordinátája.

TEXT (VDI 8)

Kiír egy megadott szöveget a grafikus képernyőre.

Deklaráció C-ben:

```
void v_gtext(handle,x,y,string)
  int handle,x,y;
  char *string;
{
  int i;
  ptsin[0]=x;
  ptsin[1]=y;
  i=0;
  while(intin[i++]=*string++);
  contrl[0]=8;
  contrl[1]=1;
  contrl[3]=-i;
  contrl[6]=handle;
  vdi( );
}
```

Meghívás C-ből:

```
v_gtext(handle,x,y,string);
```

Paraméretek:

x: a szöveg X koordinátája;
y: a szöveg Y koordinátája;
string: a kiírandó szöveg mutatója.

FILLED AREA (VDI 9)

Kiszínezi az attribútumfüggvényekkel beállított paramétereknek megfelelően (szín, minta stb.) a megadott sokszöget.

Deklaráció C-ben:

```
void v_fillarea(handle,n,pxyarray)
  int handle,n,pxyarray[ ];
{
  pioff=pxyarray;
  contrl[0]=9;
  contrl[1]=n;
  contrl[3]=0;
  contrl[6]=handle;
  vdi( );
  pioff=ptsin;
}
```

Meghívás C-ből:

```
v_fillarea(handle,n,pxyarray);
```

Paraméterek:
pxyarray[0]: az 1. pont X koordinátája;
pxyarray[1]: az 1. pont Y koordinátája;
pxyarray[2]: a 2. pont X koordinátája;
pxyarray[3]: a 2. pont Y koordinátája;
.
.
.
pxarray[2n-2]: az n. pont X koordinátája;
pxarray[2n-1]: az n. pont Y koordinátája.

Az alaprajzoló függvényeket a VDI 11-es operációs kód különböző alkódjaival érhetjük el. Ezek a GDP (Generalized Drawing Primitive) függvények.

BAR (VDI 11, GDP1)

Kitöltött téglalapot rajzol.

Deklaráció C-ben:

```
void v_bar(handle,pxyarray)
  int handle,pxyarray[ ];
{
  pioff=pxyarray;
  contrl[0]=11;
  contrl[1]=2;
  contrl[3]=0;
  contrl[5]=1;
  contrl[6]=handle;
  vdi( );
  pioff=ptsin;
}
```

Meghívás C-ből:

```
v_bar(handle,pxyarray);
```

Paraméterek:
pxyarray[0]: sarokpont X koordinátája;
pxyarray[1]: sarokpont Y koordinátája;
pxyarray[2]: átlós sarokpont X koordinátája;
pxyarray[3]: átlós sarokpont Y koordinátája;

ARC (VDI 11, GDP 2)

Körívrajzoló szubrutin. A fokokat 1/10 fokokban kell megadni.

Deklaráció C-ben:

```
void         v_arc(handle,x,y,radius,begang,endang)
  int handle,begang,endang,x,y,radius;
{
  ptsin[0]=x;
  ptsin[1]=y;
  ptsin[2]=ptsin[3]=ptsin[4]=ptsin[5]
=0;
  ptsin[6]=radius;
  ptsin[7]=0;
  intin[0]=begang;
  intin[1]=endang;
  contrl[0]=11;
  contrl[1]=4;
  contrl[3]=contrl[5]=2;
  contrl[6]=handle;
  vdi( );
}
```

Meghívása C-ből:

```
v_arc(handle,x,y,radius,begang,
endang);
```

Paraméterek:
began: kezdőszög;
endang: végszög;
x: középpont X koordinátája;
y: középpont Y koordinátája;
radius: sugár;

PIE (VDI 11, GDP 3)

Az attribútumfüggvényekkel beállított paraméterű kitöltött körívet rajzol. A szögeket 1/10 fokokban kell megadni.

A deklaráció és a paraméterek ugyanazok, mint az ARC (VDI 11, GDP 2) esetében.

Meghívása C-ből:
v_pieslice(handle,x,y,radius,begang, endang);

CIRCLE (VDI 11, GDP 4)

Körívet rajzol.

Deklarációja C-ben:

```
void v_circle(handle,x,y,radius)
  int handle,x,y,radius;
{
  ptsin[0]=x;
  ptsin[1]=y;
  ptsin[2]=ptsin[3]=0;
  ptsin[4]=radius;
  ptsin[5]=0;
  contrl[0]=11;
  contrl[1]=3;
  contrl[3]=0;
  contrl[5]=4;
  contrl[6]=handle;
  vdi( );
}
```

Meghívása C-ből:
v_circle(handle,x,y,radius);

Paraméterek:
x: középpont X koordinátája;
y: középpont Y koordinátája;
radius: sugár.

ELLIPSE (VDI 11, GDP 5)

Az attribútumfüggvényekkel előre beállított paraméterű kiszínezett ellipszist rajzol.

Deklaráció C-ben:

```
void v_ellipse(handle,x,y,xradius,yradius)
  int handle,x,y,xradius,yradius;
{
  ptsin[0]=x;
  ptsin[1]=y;
  ptsin[2]=xradius;
  ptsin[3]=yradius;
  contrl[0]=11;
  contrl[1]=2;
  contrl[3]=0;
  contrl[5]=5;
  contrl[6]=handle;
  vdi( );
}
```

Meghívása C-ből:
v_ellipse(handle,x,y,xradius,yradius);

Paraméterek:
x: középpont X koordinátája; y: középpont Y koordinátája; xradius: az ellipszis x-tengelye; yradius: az ellipszis y-tengelye.

ELLIPTICAL ARC (VDI 11, GDP6)

Ellipszisívet rajzol, a szögértékeket 1/10 fokokban kell megadni.

Deklaráció C-ben:

```
void v_ellarc(handle,x,y,xradius,yradius,begang,endang)
  int handle,begang,endang,x,y,xradius,yradius;
{
  ptsin[0]=x;
  ptsin[1]=y;
  ptsin[2]=xradius;
  ptsin[3]=yradius;
  contrl[0]=11;
  contrl[1]=2;
  contrl[3]=0;
  contrl[5]=5;
  contrl[6]=handle;
  vdi( );
}
```

Meghívása C-ből:
v_ellarc(handle,x,y,xradius,yradius,begang,endang);

Paraméterek:
began: kezdőszög;
endang: végszög;
x: középpont X koordinátája;
y: középpont Y koordinátája;
xradius: x-tengely;
yradius: y-tengely.

ELLIPTICAL PIE (VDI 11, GDP 7)

Kiszínezett ellipsziscikket rajzol. A szögértékeket 1/10 fokokban kell megadni.

A deklaráció és a paraméterek jelentése azonos az ELLIPTICAL ARC (v_ellarc) szubrutinnal, meghívása C-ből:
v_ellpie(handle,x,y,xradius,yradius,begang,endang);

ROUNDED RECTANGLE (VDI 11, GDP 8)

Lekerekített sarkú téglalapot rajzol.

Deklaráció C-ben:

```
void v_rbox(handle,xyarray)
  int handle,xyarray[ ];
{
  pioff=pxyarray;
  contrl[0]=11;
  contrl[1]=2;
  contrl[3]=0;
  contrl[5]=8;
  contrl[6]=handle;
  vdi( );
  pioff=ptsin;
}
```

Meghívása C-ből:

v_rbox(handle,xyarray[ ]);

Paraméterek:
pxyarray[0]: sarokpont X koordinátája;
pxyarray[1]: sarokpont Y koordinátája;
pxyarray[2]: átlós sarokpont X koordinátája;
pxyarray[3]: átlós sarokpont Y koordinátája;

Kovács P. Attila

A programozási nyelvek világa

A kalandozás folytatódik

És mondá az Úr:
„Íme itt egy nép, és mindannyian egy nyelvet beszélnek,
de ez még csak a kezdet."

Teremtés Könyve 11.6

A 60-as évek végének nagy szoftverválsága a programokkal és a programozási nyelvekkel szemben újabb követelményeket támasztott. Ilyen volt például a strukturált programozás nyelvi támogatása, az adatstruktúrák fontosságának felismerése, a modern szoftvertervezési metódusok támogatása és a jobb vezérlési struktúrák kialakítása.

Ortogonalitás és szimuláció

Az új követelményeken alapuló nyelvek közül a legismertebb az Algol 68, a SIMULA 67 és a Pascal. Az Algol 68 nagyon gondosan tervezett programozási nyelv, többek között itt vezették be az ortogonalitás elvét: a nyelv olyan alapvető elemeket tartalmazott, amelyek önmagukban állnak, és nem függnek a nyelv más elemeitől. Az Algol 68 azonban bonyolultsága miatt nem terjedt el.

Kortársa, a SIMULA 67 Algolhoz hasonló nyelv, elsősorban szimulációs alkalmazások számára fejlesztették ki. A SIMULA 67 legfontosabb tulajdonsága az osztály (class) fogalom volt, ami az adatabsztrakció őse. Az osztály lehetővé teszi, hogy a programozó a deklarációkat és eljárásokat egyetlen egységbe foglalja.

A Pascalt, a legelterjedtebb programozási nyelv büszke címére igényt tartó egyik aspiránst, a XVII. századi francia matematikusról, Blaise Pascalról nevezték el. A nyelvet Niklaus Wirth, a zürichi Szövetségi Műszaki Egyetem professzora tervezte 1968-ban. Az első Pascal-fordítóra 1970-ig kellett várni, de a 70-es évek végére már több mint 110 különböző Pascal-dialektus létezett, és mára a Pascal lett a számítástechnikai oktatás alapnyelve. A Pascal utódai, a Modula-2, Modula-3 és az Oberon az objektumorientált nyelvekkel együtt a 90-es évek felfedezettjei.

A vetélytárs

A másik aspiráns a legelterjedtebb programozási nyelv címre a C, amelyet Dennis M. Ritchie, a Bell Laboratórium munkatársa tervezett és implementált 1972-ben a UNIX operációs rendszer programnyelveként. A C, akárcsak a Pascal, az Algol nyelv leszármazottja. A C nyelv közkinccsé tétele 1978-ra datálódik, amikor megjelent Brian Kernighan és Dennis Ritchie könyve, a The C Programming Language, amely útmutatóul szolgált a C nyelv implementálásához UNIX-tól eltérő környezetben. A mikro- és személyi számítógépes implementációk és a UNIX operációs rendszer elterjedése nagyban hozzájárult a C sikeréhez. A C méltó utóda a 90-es évekre az objektumorientált kiterjesztést tartalmazó C++ nyelv.

Absztrakció és párhuzamosság

A 70-es években a programozási nyelvek két fő irányban fejlődtek, az adatabsztrakció és a párhuzamosság irányában. Az adatabsztrakció elve az, hogy adatstruktúrákat a rajtuk végezhető műveletekkel definiálunk.

A SIMULA osztálya egy lehetséges módszer az adatabsztrakció megvalósítására.

Ezt a fogalmat más nyelvek, például az EUCLID, a CLU és a GYPSY tovább finomították. Sőt, ez az elv általánosodott később az objektumorientált programozási nyelvekben, például a Smalltalkban.

A Smalltalkban minden objektum, és az objektumok csak üzenetekkel kommunikálnak. Dijkstra és Hoare munkáira támaszkodva olyan új programozási nyelveket is terveztek, amelyek a párhuzamos feldolgozás és a valós idejű rendszerek programozási problémáival foglalkoztak. Ilyen Brinch Hansen Concurrent Pascalja, Wirth Modulája vagy a Concurrent C nyelv.

A 80-as években elsősorban a korábbi nyelvek hiányosságait kiküszöbölő vagy kijavító és a legújabb programozástechnikai és elméleti elemeket tartalmazó nyelveket hoztak létre. Ilyenek például a Smalltalk, a Modula-2, a Modula-3, az Oberon, az Object Pascal, a C++ és a negyedik generációs programozási nyelvek (4GL) egész sora.

A programozási nyelvek egy másik fejlődési iránya az alkalmazásspecifikus programozási nyelvek és eszközök megjelenése. Ezek az elsősorban programozási és problémamegoldási segédeszköznek tekinthető programrendszerek egy-egy speciális alkalmazási területtel kapcsolatos feladatok számítógéppel támogatott megoldását teszik lehetővé. Ezek a segédeszközök meglehetősen széles skálán mozognak, kezdve a célorientált programozási nyelvektől (például adatbázis-kezelés) egészen az interaktív, felhasználó- és nem programozóorientált rendszerekig, mint a legtöbb táblázatkezelő, adatbázis-alkalmazási és a CAx (Computer Aided anything = számítógéppel segített bármi) rendszerek. Persze ezeknek a specializált nyelveknek (az egyéb programozási segédeszközökről most nem is szólva) az egyik problémája az, hogy minél közelebb kerülnek egy felhasználói csoport nézőpontjához, annál inkább távolodnak mások nézőpontjától, igényeitől. Egy táblázatkezelő program nem a legalkalmasabb eszköz mondjuk egy repülésirányító rendszer megírásához.

Reszkess, fordító!

Az Ada szinte minden korábbi nyelvben fellelhető ötletet tartalmaz (nevét Ada Augusta Byron, Lovlace grófnéről, az első női programozóról kapta). Ezt a nyelvet az USA védelmi minisztériumának kezdeményezésére tervezték. Az Ada Pascalon alapuló programozási nyelv, mely numerikus számításhoz, párhuzamos programozáshoz, adatabsztrakcióhoz és kivételkezeléshez szükséges tulajdonságokkal rendelkezik. Sokak szerint az Adát a fordítókészítők ellen eszelték ki, mivel egy ilyen

komplex rendszer implementálása egyáltalán nem könnyű feladat. Igaz, hogy több millió soros rendszer esetén már bebizonyította létjogosultságát, és a nyelvet támogató szervezet sem hagyható figyelmen kívül, de széles körű elterjedése még kérdéses.

Egy program kifejlesztése több fázisból áll. Az első lépés egy specifikáció, a rendszerterv kialakítása, amelynek célja a program által ellátandó feladatok és a program működésének meghatározása minden lehetséges körülményre. Kis programok esetében a tervezési fázis nem annyira szükséges, nagy rendszereknél azonban a jól előkészített tervezés elengedhetetlen: a tervezés és előkészítés tovább tart, mint maga a kódolás. A következő lépés a módszerek és algoritmusok kiválasztása és megalkotása. Ezek után következik a program kódolása, azaz a probléma megoldására született program elkészítése valamilyen programozási nyelven.

A számítógépprogram formájában megfogalmazott megoldást még át kell alakítani a program futtatására kiszemelt számítógép gépi nyelvére. A gépi nyelv nem probléma-, de még csak nem is programozóorientált. A gépi nyelvű program nem más, mint számok (vagy inkább kódok) hosszú sorozata, amelyek a számítógép számára érthető sorrendben követik egymást. Egy magas szintű programozási nyelven írt program (a forrásprogram vagy forráskód) átalakítását a gépi nyelvű megfelelőjére (a tárgy- vagy programkódra) egy speciális program, a fordító (compiler) végzi. Egy adott programozási nyelvnek több fordítója is létezhet akár egy adott, akár több eltérő gépre.

A fordítási fázisban a nyelv szintaxisának és szemantikájának ellentmondó hibákra derül fény. Minden nyelv alapja a szintaktikai és szemantikai szabályok meghatározása. A szintaxis rögzíti, hogy milyen karakterláncok (jelkészletek) alkothatják a nyelv jól megírt programjait, és milyenek nem. A szemantika pedig az egyes programok jelentését adja meg. Amikor egy szöveget olvasunk, akár könyvet, akár programot, a megértés első lépése az, hogy minden egyes elemnek megállapítjuk a formáját és jelentését. Ezután képezzük az egyes betűkből a magasabb szintű szavakat. A fordító hasonlóan jár el. A forráskódú programból szimbólumokat képez. Ez a folyamat a lexikális elemzés. A fordítónak kell gondoskodnia arról is, hogy a generált gépi kód lehetőleg maximálisan optimalizált legyen.

A programfejlesztés során nagy gondot és sok időt kell fordítani a program tesztelésére. Különböző tesztelési esetek létrehozásával próbálják a programban megbújó logikai hibákat és a specifikációtól eltérő viselkedést detektálni. Ez a hibakeresési fázis a „tetvészés" (debug).

Programot írni már könnyű!

Az utolsó fázis a program üzembe helyezésével és karbantartásával kapcsolatos tennivalókat foglalja magában.

Jól látható, hogy bonyolult rendszerek esetén maga a programírás a feladat megoldásának csak kis részét jelenti.

A mágneslemez mellékleten található prg-00.lst példái a programozás világába való bevezetést segítik.

Bemelegítőnek ennyi elég, folytatásként a jövő hónapban az Assembly-programozás területére kalandozunk.

Villányi László

Kívül a Clipperen

Elérkeztünk az új Clipperről szóló sorozatunk utolsó (negyedik) részéhez.
Ami lényegeset a témáról el lehetett (és kellett) mondani, azt megírtuk.
Szándékosan nem boncolgattuk a nyelvi elemeket, mert ezeket a kézikönyvek amúgy is részletesen tárgyalják.
Az újdonságokat fontosságuk szerint mutattuk be.
Nem véletlen, hogy az objektumokról egy egész cikk szólt, a PE nevű szövegszerkesztőről pedig csak egy bővített mondat...
Nos, ami mostanra maradt:

Az előző rész teljes egészében az új változótípusról, az objektumról szólt. Az ott leírtak máig is érvényesek, de van még valami, ami ha nem is szorosan, de a témához tartozik. Egy olyan dologról van szó, ami kezdettől fogva jellemzi a Clipper-változatokat, ez pedig az extend rendszer. Ez a függvénygyűjtemény paramétereken keresztüli összeköttetést biztosít a C és az Assembler nyelvvel, és természetesen ezek által még sok más nyelvvel is. Ha ragaszkodunk az objektumok használatához, megtehetjük ezt, mondjuk, C++ forrásnyelven, és hozzászerkeszthetjük Clipper programunkhoz.

Sajátom-e a sajátom?

Mint már az első részben is szó volt róla, egy teljesen legális lehetőséggel élve tetszésünk szerint átdefiniálhatjuk a nyelv egész parancskészletét (STD.CH, #command, #translate).

Az új parancsok kifejtésében használhatjuk a rendszer teljes függvénykészletét, beleértve az aláhúzás karakterrel kezdődő ún. belső függvényeket is.

Itt fel kell fedeznünk azonban még egy lehetőséget, nem is akármilyet. A parancskészlet átdefiniálásakor (megint csak természetesen) használhatunk saját függvényeket is.

Olyan saját függvényeket, amelyeket az extend rendszer segítségével C++ (vagy bármi más) forrásnyelven írtunk meg, és ebből adódóan például hemzseghetnek benne az objektumok, vagy kezelheti közvetlenül a monitort (a VGA-t is, grafikusan is), a nyomtatót, vagy uram bocsá' a merevlemezt.

Ilyen külső függvények megírásához igyekszik segítséget nyújtani az extend rendszer, amelynek minden funkciója az alapkönyvtárban (Clipper.LIB) van. A külső nyelvek felé (C és Assembler) az EXT *.* nevű fájlok tartalmazzák a definíciókat. Ezek a funkciók 3 + 1 jól elkülöníthető csoportra oszthatók.

Az első csoport a főprogramtól kapott paraméterek átvételére szolgál. Lehetőségünk van karakterstring, numerikus (int, long, double) tömb-, logikai és dátumparaméterek átvételére.

A második csoport a függvényérték visszaadására szolgál. A visszaadható típusok köre megegyezik az átvehetőkkel, kivéve a tömböt. A hossz szerinti stringvisszaadást sem használhatjuk tömbcím visszaadására, mert a Clipper a tömbelemeket speciális vezérlőkarakterekkel megtoldva tárolja.

A harmadik (egyébként új) csoport a paraméterként kapott változók értékének megváltoztatására szolgál. Itt lehetőségünk van beleturkálni a Clipper változóiba, még a tömbökbe is. Egy dologra azonban oda kell figyelni. Ha a paramétereket nem cím, hanem érték szerint adtuk át, akkor csak a változók másolatait tudjuk átírni, és ezek a saját függvényre nézve lokálisak, tehát a Returnnel leépülnek.

A negyedik csoportba három funkció tartozik, ezek a Clipper-heap-ből foglalnak le, illetve szabadítanak fel memóriát. Érdekes a két memóriaallokáló függvény közti különbség, tudniillik ha az allokáció sikertelen, akkor az egyik NULL pointerrel tér vissza, a másik visszatérés helyett runtime hibát generál.

Az extend rendszer nem támogat más forrásnyelvet, mint a C és az Assembler, de ennek ellenére megvan a lehetőség például Pascal típusú hívási konvenciójú felhasználói függvények írására is. Az egyetlen kikerülhetetlen szabály az, hogy a Clipperhez való csatlakozáskor a C nyelv LARGE modell szerinti hívási és paraméterátadási előírásait kell alkalmazni.

Új függvények halmaza

Habár a sorozatnak nem célja nyelvi szintre ereszkedve boncolgatni az új Clipper-verziót, semmiképpen sem tartom károsnak, ha néhány érdekes dolgot innen is kiemelünk. Ez még nem jelenti azt, hogy alkalmazástechnikai mélységekben kell gondolkodnunk, hanem csupán annyit, hogy belepillantunk a The Guide To Clipper című Norton-adatbázisba.

Az első felületes átnézésre találunk 15 új függvényt, amelyeket a későbbiekben kettő + egy csoportba fogunk besorolni. Találunk ezenkívül 22 olyan függvényt és utasítást, amelyek Nantucket-terminológiával élve „elavultak". Ez utóbbiakat az új verzió csak kompatibilitási okokból őrizte meg, nagy részüket a preprocesszor az STD.CH irányelvei szerint egyszerűen kigyomlálja a forráskódból. Ha új programot írunk, ezeket az utasításokat lehetőleg már ne használjuk, mert mindegyiknek van „modernebb" megfelelője. További vizsgálódással találunk még három olyan függvényt, amelyekről a Nantucket azt állítja, hogy jelentősen megváltoztak.

Minket most leginkább az új függvények halmaza érdekel. Ami először szembetűnik, az az, hogy minden új lehetőséget függvényeken keresztül tudunk elérni. Még véletlenül sincs egyetlen új utasítás sem.

Ez az új Clipper függvényorientált jellegéből adódik.

Most nézzük az új függvények beígért csoportjait! Az első, a legtöbb függvényt tartalmazó csoport a képernyőt, az stdout, illetve az stderr perifé-

riát kezeli. Ezekkel a függvényekkel sokkal jobban kézben tudjuk tartani programunk outputjait, mint ahogy azt eddig megszokhattuk. A kiíratásra szolgáló funkciók három kimeneti eszközt különböztetnek meg egymástól. Lehetőségünk van az outputot a képernyőre, az stderr-re, az stdout-ra, vagy éppen a Clipper kurrens device meghajtójára küldeni. Az utolsó kettő nem feltétlenül egyezik meg. Ebben a csoportban találunk még három, kifejezetten a képernyővezérlést manipuláló funkciót. Ezekkel megszüntethetjük a havazást a régi típusú CGA-kártyás gépeken is, megszabhatjuk, hogy az attribútumbájt legfelső bitje a villogásra vagy a háttérintenzitásra vonatkozzon-e, valamint a hardverfeltételek megléte esetén kiválaszthatunk például: 40x25, 80x25, 132x25, 132x60 stb. szöveges megjelenítési módokat. Az új rendszer is fájdalmasan nélkülöz mindenféle grafikus lehetőséget.

Az új funkciók második csoportját a kódblokkgeneráló függvények alkotják. Ezek tulajdonképpen egy-egy függvény címét adják vissza, amelyeket később saját paramétereinkkel hívhatunk meg.

Az utolsó (egyéb) csoportba két új függvény került. Az egyik a get objektumok kezelésében játszik szerepet. Visszatérési értéke az éppen bekérés alatt álló get objektum címe. A másik egy sokkal fontosabb dolog, éspedig a BREAK( ). Az új verzióban lehetőségünk van egy BEGINSEQUENCE és ENDSEQUENCE pároson belül létrehozni egy olyan programrészt, amelyre csak egy BREAK( ) utasításon keresztül kerülhet a vezérlés. Ennek a résznek a RECOVER( ) jelöli az elejét, és az ENDSEQUENCE a végét. Ha a program BREAK( ) nélkül érné el a RECOVER( )-t, úgy az említett programrész átlépésre kerül. Fontos tudnunk, hogy a BREAK( )-nek nem kötelezően ugyanazon függvényben kell lennie, mint a SEQUENCE-nek. Ezt a lehetőséget kihasználja az előző részhez mellékelt példaprogram is.

Ezzel a négyrészes sorozat végére értünk. Még annyit kívánok megjegyezni, hogy bátran ajánlom az új Clipper-változatot minden olyan kedves olvasónak, aki új programokat fejleszt, és legalább 80286-os, 16 MHz-es gépe van, 1 Mbájt RAM-mal. Már meglévő programok újrafordítására, esetleg „feljavítására" nemigen alkalmas a rendszer, az említett konfigurációnál lassabb gépeken pedig olyan bosszantóan lassú, hogy nem merem ajánlani a használatát.

Fridl György

Modula elemi típusok 2.

Folytatva a júliusi számban megkezdett ismerkedést a Modula-2 elemi típusaival, ebben a részben a BOOLEAN, CHAR és BITSET típusokon a sor.

Mindennapi életünkben sok dolgot fejezhetünk ki két, egymást kölcsönösen kizáró állítással. Például vagy otthon vagyok, vagy nem; vagy megírom a soron következő Modula-cikket, vagy nem; vagy van elég pénzem egy Ferrarira, vagy nem (ez mellesleg csütörtökön dől el). Ezért aztán a valós élet problémáit megoldó programozási nyelveknek is rendelkezniük kell olyan típussal, amely éppen két értéket vehet fel. Ezek az úgynevezett logikai vagy Boole-változók (George Boole matematikus után).

A BOOLEAN típus

A Modula-2 előre definiált BOOLEAN típusa ilyen változók deklarálását teszi lehetővé. Egy BOOLEAN típusú változó értéke logikai igaz vagy hamis lehet. Ezt a két értéket a Modulában a TRUE (igaz) és FALSE (hamis) standard azonosítók jelölik. A logikai változók azonosítói sokszor melléknevek, ilyenkor a TRUE, illetve FALSE értékek a jelölt tulajdonság meglétét vagy hiányát fejezik ki. A logikai változók a Modula logikai operátoraival együtt logikai kifejezéseket alkotnak. Ezek a logikai operátorok az AND (ÉS), az OR (VAGY) és a NOT (NEM). Az AND-re használható az „&", míg a NOT-ra a „~" jel is. A logikai operátorok jelentése:

a AND b = „a is és b is igaz (TRUE)"

a OR b = „vagy a, vagy b, vagy mindkettő igaz (TRUE)"

NOT a = „a hamis (FALSE)"

OR szokásos elnevezése még az „inclusive OR", ami arra utal, hogy a logikai művelet eredménye abban az esetben is igaz, ha mindkét operandus értéke igaz. Ezzel ellentétben a logikai kizáró VAGY operátor (exclusive OR, EXOR vagy XOR) értéke csak akkor igaz, ha vagy az egyik, vagy a másik operandus értéke igaz. Ha mindkettő értéke igaz, az eredmény hamis lesz. A logikai kizáró vagy azonban nem alapművelet, mivel a korábbi három művelet segítségével előállítható (A XOR B = (A OR B) AND (NOT (A AND B))).

A Modula-2-ben a logikai operátorok definíciója valamelyest eltér a logikai definíciótól, de az eredmény a várt lesz.

a AND b = IF a THEN b ELSE FALSE

a OR b = IF a THEN TRUE ELSE b

Ez a definíció azt tükrözi, hogy a művelet kiértékelésekor az első operandus értékéből megállapítható az eredmény, a második operandus kiértékelésére nincs szükség. Ennek a definíciónak az egyik leglényegesebb következménye az, hogy a logikai kifejezések értéke azokban az esetekben is jól definiált, amikor a második operandus értéke nem definiált. Ez különösen a Pascalban programozók számára jelent újdonságot, és megegyezik a C-beli kiértékeléssel. Ennek az a kihatása, hogy az operandusok sorrendje szignifikáns lehet egy adott kifejezésben.

Példa: az IF (X # 0.0) AND (Y / X > 4.0) THEN … kifejezésben az osztás csak akkor lesz kiértékelve, ha X értéke nem nulla.

A logikai kifejezéseket egyszerű transzformációs szabályok alapján egyszerűsíthetjük. A legismertebb ilyen transzformációs szabályok az úgynevezett de Morgan-törvények, amelyeket az alábbi egyenlőségek definiálnak:

(NOT A) AND (NOT B) AND … = NOT (A OR B OR …)

(NOT A) OR (NOT B) OR … = NOT (A AND B AND …)

A Modula-2 relációi (összehasonlító műveletei) BOOLEAN típusú eredményt szolgáltatnak, azaz az eredmény TRUE, ha a reláció fennáll, és FALSE, ha nem. Például:

7 = 34 FALSE

7 < 34 TRUE

45.89 >= 6.78 FALSE

A relációk szintaktikailag a kifejezések osztályába tartoznak, és az összehasonlítandók úgynevezett egyszerű ki-

fejezések. Az összehasonlítás eredményének típusa BOOLEAN, és így használható az IF, WHILE és REPEAT vezérlési struktúrákban. (A kifejezésekről és utasításokról később részletesen lesz szó.)

Formálisan:

$ Kifejezés := EgyszerűKifejezés [reláció EgyszerűKifejezés].

$ reláció := „=" | „#" | „<>" | „<" | „<=" | „>" | „>=" | „IN".

Megjegyzés: „#" és „<>" jelentése „nem egyenlő".

A BOOLEAN értékekkel is végezhetünk összehasonlításokat, és nem csak egyenlőség eldöntésére (pl. FALSE < TRUE, TRUE = FALSE).

Ahogy azt már korábban láttuk, a Modulában egy operátor operandusainak azonos típusúaknak kell lenniük, és ez a relációs operátorokra is vonatkozik . Így a következő relációk hibásak:

```
1 = TRUE
5 = 5.0
A+B = C OR D
6.5 > 2
0 # TRUE
```

Ugyancsak hibás például az x <= y < z kifejezés, de átalakítható (x <= y) & (y < z) formára. Az alábbiak már helyes logikai kifejezések:

```
x = y
(x > p) OR (p > k+z)
NOT B OR C
I+J < K*D
P OR T = (I < J)
```

Megjegyzés: Bár az l = TRUE és k = FALSE típusú kifejezések megengedettek, mégis az l, illetve NOT k (~k) jelölésmód használata a gyakoribb, és programozástechnikailag is elegánsabb. Ugyancsak elegánsabb megoldás a logikai változók értékadása relációs műveletek segítségével.

Példa:

```
IF string[i] = 0C THEN
eol := TRUE ;
ELSE
eol := FALSE ;
END ;
```

helyett

```
eol := string[i] = 0C ;
```

A BOOLEAN egyébként formálisan egy előre definiált felsorolástípus az alábbi deklarációval:

```
TYPE BOOLEAN = (FALSE, TRUE) ;
```

Így TRUE és FALSE előre deklarált konstans azonosítók (a felsorolástípusokról majd még bővebben lesz szó).

Példák: A Boole-algebrával és a BOOLEAN típusokkal kapcsolatos példák az M2-03.LST fájl 1-4.listájában és az EX03-01.MOD programban találhatók.

A CHAR típus

A számítástechnikai alkalmazások nem csak numerikus feladatok megoldásából állnak. A szövegfeldolgozás és -kezelés alapvető fontosságú olyan feladatok elvégzéséhez, mint például editorok, szövegszerkesztők, irodaautomatizálás, vagy hogy szűkebb témánknál maradjunk, a fordítók, amelyek tulajdonképpen speciális célú szövegfeldolgozók. Ezenfelül szinte minden számítógépes rendszer vagy program a környezetével valamilyen ki- és beviteli eszköz segítségével kommunikál, amikor is a felhasználói kommunikációnak az ember számára is érthető szöveges formában kell lezajlania. Ebből adódóan szinte nincs is olyan számítógépes program, amely ne tartalmazna valamilyen szövegfeldolgozási és értelmezési feladatot.

A számítógép által előállított vagy feldolgozott szöveg elemei (szavak, jelképek) karakterek egy meghatározott halmazából valók. A szöveg alapja a karakter, és a Modulában a karaktereket a CHAR típus segítségével lehet kezelni. Az adott számítógépen alkalmazott karakterhalmaz vagy karakterkészlet adja a a CHAR típus értéktartományát. Sajnos a különböző típusú számítógépek nem mindig azonos karakterhalmazt használnak, ami a közöttük lévő kommunikációt (a programok és adatok cseréjét) megnehezíti. Például a „+" karakter minden bizonnyal szerepel minden számítógépen, de mondjuk a „%" vagy az „á" szinte biztos, hogy nem.

Szerencsére létezik egy nemzetközi szabvány karakterkészlet, az úgynevezett ISO. Az ISO-szabvány egy 128 karakterből álló halmazt definiál, amelyek közül 33 úgynevezett vezérlőkarakter. A fennmaradó 95 elem megjeleníthető karakter. Ez a halmaz rendezett, és mindegyik karakternek van egy meghatározott helye, illetve sorszáma. Például az „A" betű a 66. karakter, a sorszáma 65. Az ISO-szabvány néhány karakter helyét szabadon hagyta, ezeket az egyes országok nemzeti karakterkészletének megfelelő karakterekkel lehet feltölteni a nemzeti szabványoknak megfelelően. A legelterjedtebb az amerikai nemzeti szabvány, röviden ASCII (American Standard Code for Information Interchange). A számunkra érdekes PC-kompatibilis környezetben ennek egy 8 bites karakterkódos változata, a kibővített (extended) ASCII karakterkészlet használatos, amely 256 karaktert definiál. A legfontosabb (?) nemzeti karakterkészleteket a PC-ken a COUNTRY és más DOS-parancsok segítségével érhetjük el. A nemzeti karakterkészlet nem azonos a 256 elemű tábla használatával (pl. CWI SZKI-kódkészlet), hanem az „alsó" 128 karakter közül a szabványban definiált „szabad" helyek országonként eltérő alkalmazását jelenti. Például a font vagy a dollár jelkép, vagy egy „napocska" karakter (angol, amerikai, svéd), vagy az ékezetes betűknek a különböző zárójelek helyén való elhelyezése (svéd, német).

A CHAR típusú állandókat idézőjel vagy dupla idézőjel közé tett karakterekkel jelöljük. Karakterértékeket CHAR típusú változókhoz rendelhetünk hozzá, de ezeket az értékeket nem használhatjuk aritmetikai műveletekben. Aritmetikai utasításokat azonban végezhetünk a karakterek sorszámaival, amelyeket az ORD(ch) típus konverziós függvény segíségével kaphatunk meg. Fordítva egy n sorszámú karaktert a CHR(n) típus konverziós függvény segítségével állíthatunk elő. A két komplemens függvény közötti összefüggés:

CHR(ORD(ch)) = ch és ORD(CHR(n)) = n, ahol: 0 <= n < 128|256.

Így lehetőség van a ch számjeggyel megjelenített szám előállítására ORD (ch)-ORD("0") alakban, illetve az n számot jelképező számjegy előállítására CHR(n+ORD("0"). Ez utóbbi két egyenlet abból adódik, hogy az az ISO-karakterhalmazban a 10 számjegy egymás mellett szerepel. Ezeket az összefüggéseket tipikusan számjegyeknek számokká, illetve számoknak számjegyekké alakításakor használják. Mivel a karaktertípus sorszámozva van, az INC és DEC standard eljárások is használhatók.

Az ISO-készlet első 32 és 128. karaktere: vezérlő- vagy kontrollkarakterek. A vezérlőkarakterek különböző célokra valók: leginkább a különböző eszközök funkcióinak vezérlésére (printer, terminál, modem), illetve szövegekben sorelválasztásra és strukturálásra (sorvége, lapvége stb.). A kontrollkaraktereket általában rövidített nevükkel szokás jelölni, amely a jelentésükre utal. Ezeknek a kódoknak az elnevezései a soros vonali kommunikációban betöltött szerepükre utalnak, más eszközök vagy alkalmazások számára azonban a kódok jelentése vagy hatása eltérő. A kódok jelentését a karakterkód nem tartalmazza, csak az adott eszköz számára definiálja a működést. Egy, a PC-s környezetben elterjedt másfajta jelölés és elnevezésmód az első 32 karakternek a nagybetűkódokhoz viszonyított helyzetét, illetve a

PC-billentyűzetről való elérhetőségét veszi alapul (pl. soh=kontroll A=^A=CtrlA). Ez az elnevezésmód már nem köti össze a kódot az esetleges vezérlési funkcióval. A kontrollkarakterek neveit és kódját a Modula-2 ASCII nevű standard könyvtármodulja tartalmazza.

A nagygépes rendszerekben régebben általános volt az oktális számrendszer használata, ezért a karakterkódokat is oktális formában adták meg. A Modula is ezt a jelölésmódot követi a karakterkódkonstansok másik megadási módjaként. Hogy a nem nyomtatható karakterek is megkülönböztethetők legyenek, a Modulában a karaktereket oktális sorszámuk és egy „C" betű is jelölheti. Például 14C egy CHAR típusú érték, amely az ff (form feed) vezérlőkaraktert jelöli, amelynek oktális sorszáma 14B. Természetesen tetszőleges karaktert helyettesíthetünk oktális kódjával (pl. 101C="A" ...).

Példák: A CHAR típussal kapcsolatos példák az M2-03.LST fájl 5-7. listájában és az EX03-02.MOD programban találhatók. A CHAR (EX03-02) program átirányított kimenete megtalálható a lemezen a char.txt fájlban. A programozás során a fájlnak az oktális karakterkód-karakter összerendelésekkor a programozás során nagy hasznát vehetjük.

A BITSET típus

A BITSET típushoz tartozó értékek: 0 és N—1 közötti egész számok halmaza, ahol N az adott számítógépre jellemző konstans, általában egy gépi szó hossza bitekben (például PC-n 16), vagy a gépi szó hosszának néhányszorosa, de ez utóbbi nem jellemző. Az ilyen típusú állandókat halmazként (l. később) jelöljük.

Példa: {5,7,11} {0} {8...15} {0...3,11,15}

Megjegyzés: Az m...n jelölés az m, m+1, ..., n—1, n rövidítése.

A BITSET típust (és általában a halmaztípusokat) a számítógép bitek halmazaként jeleníti meg, azaz a halmazra jellemző alapvető funkciónak megfelelően, miszerint egy elem egy adott halmaz része vagy nem. Például u gépi szó i-edik bitje 1, ha i u halmaz eleme, és 0, ha nem. Ez azt jelenti, hogy a magas szintű halmazabsztrakción túl, amit a Modula SET típusa nyújt, a Modula-halmazok a számítógép legkisebb egységének, a bitnek a kezelésére is alkalmasak. Az előre definiált BITSET típus lehetővé teszi egy gépi szó vagy regiszter bitenkénti kezelését. A halmazoperátorok részletes ismertetésére majd később kerül sor. Most csak azt nézzük meg, mit jelentenek ezek a BITSET, mint alacsony szintű gépközeli típusra alkalmazva:

+ halmazunió, azaz bitenkénti OR

- halmazkülönbség, azaz bitenkénti AND NOT

* halmazmetszet, azaz bitenkénti AND

/ szimmetrikus halmazkülönbség, azaz bitenkénti XOR

e IN h TRUE, ha e eleme h-nak, azaz TRUE, ha h e-edik bitje 1 INCL(halmaz,elem) elem felvétele halmazba, azaz halmaz elemedik bitjének 1-re állítása EXCL(halmaz,elem) elem törlése halmazból, azaz halmaz elemedik bitjének törlése.

A halmazoperátorok általánosságban a halmaz N elemén végzett logikai operátorként vannak implementálva. Ezért nagyon hatásosak, és végrehajtási idejük általában sokkal rövidebb, mint például az egész számok összeadásáé. Ez különösen olyan processzorokon „látványos", amelyek képesek bitek és bitcsoportok kezelésére is, mint a Motorola 680x0 sorozata, de az elemi halmazműveletek még az Intel 80x86 sorozaton is igen gyorsak.

A Modula-halmazokról és a halmazokon végezhető műveletekről később részletesen is szó lesz.

Példák: A BITSET típussal kapcsolatos példák az M2-03.LST fájl 8-9. listájában és az EX03-03.MOD programban találhatók.

E havi „utility" modulunk a hónap témájához kapcsolódva bináris-hexa, illetve hexa-bináris átalakítást végez. A program szintaktikája „binhex befájl.kit kifájl.kit kapcsoló", ahol befájl.kit az átalakítandó fájl (teljes) neve kiterjesztéssel, kifájl.kit az eredményfájl (teljes) neve kiterjesztéssel, és kapcsoló az átalakítás formáját meghatározó karakter „c" vagy „d", amelyet egy „—" vagy „/" jel előz meg.

A számítógép használata közben gyakran van szükségünk olyan fájlokra, amelyek nem tartalmaznak grafikai és vezérlőkaraktereket. E problematikus fájlok általában a bináris (.exe, .com, .bin stb. kiterjesztésű fájlok), amelyek nyomtatása, soros vonalon való elküldése, vagy akárcsak képernyőre listázása gondot okozhat. Ilyen esetekben jön jól a binhex program, amely -c (convert) kapcsolójával a bináris fájlokat kizárólag hexadecimális számokat tartalmazó fájlokká konvertálja.

Mivel ezek csak nyomtatható (hexadecimális számjegyek és sorvége jelek) karaktereket tartalmaznak, tetszőleges módon manipulálhatók, akár módosíthatók is.

A hexbin program -d (deconvert) kapcsolójával a hexadecimális fájlok bármikor átalakíthatók binárisakká. A -d kapcsoló használata előnyös lehet akkor is, amikor a külföldi számítástechnikai folyóiratokban gyakori „utility" .com fájlokat szeretnénk „magunkévá tenni". Mivel a külföldi lapok olvasói köre sem csak assembly programozókból áll, a lapok közölni szokták a .com fájl előállítását végző BASIC program listáját is. A teljes program bepötyögtetése helyett elegendő a DATA utasításhoz tartozó hexadecimális karakterpárokat begépelni egy szöveges fájlba, a .com fájllá alakítást már rábízhatjuk a hexbin programra. Ez annál is inkább célszerű a hazai viszonyokat tekintve, mivel a nálunk domináló távol-keleti PC-klónok nagy részének nincs BASIC fordítója. (A hexbin -c kapcsolója hasonló módon lehetővé teszi a DATA típusú szöveges fájl előállítását.)

A mágneslemez melléklet tartalmaz két .asc kiterjesztésű fájlt: a fastkeys.asc és numoff.asc fájlokat, amelyek .com verziója a saját autoexec.bat fájlomban található; ezenfelül egy keyclick.asc fájlt. Hogy mit csinálnak, azt mindenki megtudhatja, aki elvégzi a visszakonvertálást. Jó pötyögtetést!

Villányi László

INFORMÁCIÓKÉRÉS: 32 ▼

Hannibál azelőtt portás volt!

Amióta a számítógépek rohamos fejlődésének és széles körű alkalmazási lehetőségeinek a távlata kirajzolódott, a gépi fordítás feladatának megvalósítása is napirenden van. Noha ezen a téren jelentős eredmények is születtek, a végcél még igen távoli, hiszen maga a nyelvtudomány sem alkotta még meg mindazokat az eszközöket, amelyek révén a fordításban szerepet játszó nyelvek tulajdonságai, törvényszerűségei pontosan leírhatók volnának. Még ha igaznak bizonyulna is Noam Chomsky amerikai nyelvésznek az a feltételezése, miszerint minden (ún. külső) nyelv egy belső, univerzális grammatikán alapul, amely mintegy a génekbe van írva, és ha meglenne is e grammatika leírása, mindaz, ami még az egyes konkrét nyelvekben s az egyes nyelvi tájak, meg az egyén sajátos nyelvszokásaiban erre az alapra rárakódik, a gépi fordítás feladatát végtelenül bonyolulttá tenné. Minél magasabbra hágunk ugyanis az alaptól a nyelv felső rétegei felé, az értelmezhetőség is annál inkább homályosul a szabatosságtól az intuíció irányába.

A gépi fordítás módszereinek megalkotása közben mindenesetre az alapoktól kellene a felsőbb rétegek felé haladni, s mindenekelőtt e rétegeket kellene egymástól elválasztani. Ez a gyakorlat annyit is jelentene, hogy az egyes nyelveknek egy sor különböző szintű, leegyszerűsített változatát kellene elkülöníteni, amelyeken belül a kutató számára minden világos és meghatározott.

A feltétel — „világos és meghatározott" — annál nehezebben teljesíthető, minél magasabb szintre lépünk a jól megfogható törvényszerűségek közül a nyelvi intuíciók meg az ún. felszíni struktúra irányába.

A gépi fordítás módszereinek kutatása nemcsak önmagában hasznos, hanem serkentően hat a nyelvtudományra is. Ahhoz azonban nem kell sem nyelvtudósnak, sem számítástechnikában jártasnak lennünk, hogy belássuk: két nyelv nem keverhető össze, mint a kénpor és vaspor, mert ha mégis megkíséreljük, abból egy nyelvnek nem nevezhető, kommunikációra alkalmatlan kotyvalék keletkezik.

Valami elképesztő misztifikáció az, hogy a számítástechnika különleges terület, amelynek leírására sajátos nyelvi eszközök szükségesek, s hogy ezeket csupán az angol nyelv képes nyújtani. Semmi másról nincs szó, mint hogy az elmúlt évtizedekben a számítástechnika súlypontja az angolszász nyelvterületre esett, s esik ma is, a vele kapcsolatos fogalmak is főként ott alakultak ki, és kaptak nevet.

Azt a törekvést, hogy az új fogalmakat az egyes nyelvterületeken saját nyelvi köntösbe öltöztessék, nacionalizmusnak nyilvánítani, a helyenként kialakult kotyvalékdialektusokat pedig a szakszerűség, netán a tudományosság szükségszerűségének nevezni nem más, mint a misztifikáció vetülete, alátámasztva anyanyelvi kisebbrendűségi érzéssel.

Mi köze van a magyar számítástechnikai szaknyelv kialakulására való törekvésnek a „díszmagyarhoz", ahogy erre K. J. utal (Alaplap, 1991. áprilisi szám). Mások tevékenységével kapcsolatban emlegethető volna hasonló hangsúllyal az „orosz medve", a „lengyel sas" és a „gall kakas" is. Végül kiderülne, hogy John Bullt és a puritánok utódait kivéve mindenki bűnös, aki él.

A franciákat — nyelvvédő törvénykezésük miatt — nemzeti elfogultsággal vádolja K. J. és Eric A. Weiss is (Abacus, Vol. 4. No. 2. 30-33.) K. J. azt is bizonygatja, hogy ez nemcsak rossz, de eszetlen dolog is. Ám miért lenne a francia szaknyelv szörnyű, mint állítja? Szörnyű annak, aki nem jártas ebben a nyelvben. Aki nem ismeri pl. a francia zoológusok szaknyelvét, annak a zoológiát francia nyelven tanulmányoznia ugyancsak keserves lehet. De kínos ám a franciául gyengén tudónak egy francia regény olvasása is. Aki jól tud angolul, viszont kevéssé franciául, ne olvassa angol cikkek francia fordítását. S ha mégis ezt teszi, ne vonjon le következtetéseket a francia szaknyelv borzalmas mivoltáról, legföljebb saját nyelvtudásának gyönge állapotáról.

Az új számítástechnikai eredmények gyors befogadására igényt tartók ma még nem nélkülözhetik az angol nyelvet. Lehet azonban, hogy holnapra ugyanők a japánnal kényszerülnek megismerkedni. A leülepedett ismeretanyagnak és a megszilárdult fogalomvilágnak azonban az anyanyelv szerkezetében kell megjelennie. A francia számára a francia, a német számára a német, a magyar számára a magyar nyelv szerkezetében. A különböző népek létét és nyelveik létezését nem lehet mellékes körülményként kezelni. Egykor talán az egész világ egyetlen, redundancia nélküli és értelmezésbeli kétségeket nem hagyó nyelvet használ majd. Ha megvalósul, ez lesz a Madách által elképzelt falanszter. Addig azonban kinek-kinek a saját nyelvét a minél magasabb szintű kommunikációra alkalmas állapotban kell tartania. Ez az intellektuális igényesség sine qua nonja. Ebben az értelemben — igen — „minden áron" (by all means) nyelvünk tökéletesítésén kell dolgoznunk. Nehezen hihető, hogy angol—magyar kotyvalék használatára való törekvés az értelem számára bármivel indokolható volna.

Amikor K. J. a port helyett javasolt porta szót kifogásolja, megjegyzéséből ítélve nyilván úgy véli, hogy a „Hannibál ante portas" kifejezést „Hannibál azelőtt kapus volt"-ként kell értelmeznie. A porta latin szó, s mint ilyen honosodott meg nálunk. Eredeti értelmében kaput jelent. Beleillik a magyar hangzásvilágba. Ugyan miért kellene nekünk most az ugyanazt jelentő, de gyötörtető, a nyelvünkből kilógó port formát használnunk? (Várom a magyarázatát.) Használhatnánk persze a kapu szót is, de azt a szakma más célra, s nagyon határozott jelentéssel már lefoglalta.

Nem értem, mi baj van a sín szóval.

Azt az áramköri berendezést, amelyet az angol bus-nak nevez, nálunk már a számítástechnika megjelenése előtt évtizedekkel sín-nek nevezték. A bus a sín fogalmához semmi többletet nem ad, de ha mégis, ezt az adat hozzáfűzésével elintézhetjük: data bus = adatsín.

A kipoppoljuk vagy besztekkeljük szavakról egyáltalán nem szellemi képzeteink támadnak; az előbbihez a peepshow popóbemutatója, az utóbbihoz a bifsztekkel való tönkrezabálás képzete járul.

Az ember fogalomvilágának tágulási korszakaiban, mint a jelenben is, minden nyelvterületen kényszerű nyelvújítás folyik, s ennek nem minden mozzanata sikeres. Születnek sikertelen magyarítások is, ezek majd lassan eltűnnek. Nyelvújítás folyik az angol nyelvterületen is. Ott sem minden sikeres, de ők szabadabb lélekkel, görcsmentesen tevékenykednek, míg nálunk lépten-nyomon csomó kerül a fonálra. Reméljük, hogy a sokáig rabságban tartott magyar lélek fölszabadultával e csomók is oldódnak, s nem érzi majd senki sem gyanúsnak vagy megvetendő tevékenységnek, ha valaki anyanyelvét óvja, csinosítja, tökéletesítgeti. Hisz, mondjanak bármit is, ez a nyelv tápláló emlője volt — e században is — az emberiség annyi kiváló elméjének.

Sebestyén Béla

Felelet a Mondolatra, avagy Ceterum censeo

Annak idején a római parlamentben volt egy úr, Porcius Catónak hívták. Minden mondókáját imígyen kezdte: Ceterum censeo Carthaginem esse delandam, azaz: javaslom, Karthágót el kell pusztítani. Nos ennek az úrnak a hangvételére emlékeztet az a vita, ami a számítástechnikai kifejezések magyarítása miatt most van kibontakozóban.

Előre kell bocsátanom: nem vagyok feltétlenül a magyarítás ellen. Egyet szeretnék csupán, a kifejezések szabatosságát és kifejezésbeli elfogadhatóságát. Gondoljunk bele: a bug-ot a sznobok kivételével mindenki poloskának hívja, de a debuggert már senki sem nevezi poloskairtónak. Bezzeg az antivírus programokból antibaci lett a szakma nyelvén.

Magam rendszeresen olvasom az angol, német és néha a francia szakirodalmat. Olyan szenvedélyes hitvita azonban, ami nálunk van a magyarítás ügyében, csak a franciáknál volt. A nyelvhez lehet érzelmi alapon is állni. De ez eddig a történelemben nem vezetett jóra.

Gondoljunk csak nyelvújítás korabeli magyarításainkra. Ezek adtak sok olyan szót — főként a köznyelvnek —, ami ma is nyelvünk része. Zongorázunk és orgonazenét hallgatunk a lemezjátszóról a horgonyfedeles kerti zenepavilonban. Az azonban eszünkbe sem jut, hogy élényt szívunk a levegőből, és húgyanyt választunk ki magunkból, amikor az árnyékszéket látogatjuk meg a szünetben.

A nyelvek fejlődése sok esetben nem követi a nyelvújítók szándékait. A társadalom dönti el — bármily jónak tűnik is egy-egy szó —, hogy mikor és milyen jelentéstartalommal fogadja azt el. A számítástechnika ennél még egzaktabb követelményekkel rendelkezik. Hasonlóan a vegyészekhez és az orvosokhoz, teljesen azonos és egyértelmű dolgokat kell érteniük a szakembereknek Budapesten éppúgy, mint Szabolcsban vagy Baranyában. Az elfogadott magyar kifejezések mellett az alap — éppen nemzetközi egyértelműsége miatt — még sokáig az angol nyelv marad.

Lehet hogy Hannibál portás volt, de a gép kimenete szerintem akkor is port, legfeljebb csatlakozó marad, mint ahogy a winchester is legfeljebb merevlemez. A magyarítási kísérleteket illetően pedig türelmesen megvárom, mit vesz át belőlük a szakma. És nem kiáltok Cato nyomán a nyelvet öncélúan újítók ellen. Úgyis a közvélemény dönti el, melyik szó marad meg a köztudatban és melyik fog kiveszni. Próbálkozni viszont érdemes!

Kis János

Nyerhet egy AT-t...

...és még sok értékes tárgyat, ha augusztus 31-ig válaszol kíváncsi kérdéseinkre. De ne csak ezért küldje vissza előző (júliusi) számunkhoz mellékelt közvéleménykutató kérdőívünket, hanem mert lapunk szerkesztését a válaszok birtokában tudjuk az Ön igényeihez is jobban közelíteni.

A Cédrus irodáinak és az Alaplap szerkesztőségének postacíme változatlan (1251 Budapest, Postafiók 71), de emlékeztetjük, hogy júliustól a Floppyland kivételével valamennyi részleg új helyre költözött: 1251 Budapest XI., Karolina út 17. Telefon: 166-2111, Fax: 185-2221.

A Mikrobazár rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás kereskedelmi tevékenységet folytatóknak gépelt soronként (60 karakter) 100 Ft, másoknak az első sor 50 Ft, minden további sor 20 Ft.

Kérjük, hogy a hirdetés díját a Cédrus Rt.-nek a Budapest Banknál vezetett 380-66760 számú számlájára utalják át, vagy postautalványon a Cédrus Rt. címére (1251 Budapest XI., Karolina út 17.) fizessék be, a hátoldalán feltüntetve, hogy apróhirdetés. A befizetést igazoló szelvényt a közlendő hirdetési szöveggel együtt az Alaplap szerkesztőségéhez küldjék el: 1251 Budapest, Pf. 71.

A PC Turbo Klub tagjai ebben a rovatban 20%-os kedvezménnyel hirdethetnek!

ADOK

Enterprise-programok eladók. Válaszboríték ellenében listát küldök. 2000 program, sok kedvezmény, ajándék. Cím: Zemen László, 1104 Budapest X., Kada u. 141. fszt. 9.

Amigára eladó több mint 2000 lemez játékprogramokkal és felhasználói programokkal. 3,5"-os lemezek 380 Ft-os, 5,25"-os lemezek 750 Ft-os áron eladók. Cím: Keresztes Gábor, 1142 Budapest XIV., Laky-köz 11. Tel.: 251-2523.

Eladó alig használt **Commodore Amiga 500** egérrel, A520 típusú modulátorral, joystickkel és 40 db lemezzel. Ára: 50 000 Ft. Cím: Simon Péter, 8710 Balatonszentgyörgy, Egry J. u. 57.

Eladó **Commodore-64,** floppyvezérlővel, lemezekkel, programokkal. Tel.: Heinbach József, (06-27) 42-600.

Eladók **C-64-hez** profi hardverek: Action Replay, Page Fox, Video Digitizer, IC-teszter, EPROM-égető stb. Szuper szoftverek lemezzel (5,25"-os, DS/DD, csak 65 Ft/lemez) együtt is eladók! Kérem, küldjön válaszborítékot! Cím: Oláh Lajos, 3014 Hort, Kossuth út 147.

Eladó **Commodore-128** 1541-es floppymeghajtóval. Cím: Treszl János, 7030 Paks, Szabó E. u. 10. Tel.: (06-75) 11-128

CLIPPER 87 EXE és OVL programok visszaalakítása forrás formátumúvá – EXE/OVL/OBJ/LIB fájlok – optimalizálás LP-vel. Cím: DECOMPILER STUDIO, 6001 Kecskemét, Pf.: 298. Tel.: (06-76) 26-515.

Eredeti **lemezek** eladók: 3,5"-os DS/DD 690 Ft, 5,25"-os DS/DD 380 Ft. Cím: Beregszászi Gábor, 1025 Budapest II., Battai u. 2. Üzenet: 155-9126.

IBM XT/AT kompatibilis számítógéphez Eprom, mikroController, Prom égetésére alkalmas **fejlesztői interfészkártya** eladó. Cím: Póka László, 3301 Eger, Pf. 415.

Eladó **MPS 1230-as nyomtató** CWI/szabvány szerinti **„magyarosítása"**. Tel.: 1848-650 (este).

Eladó 1 éves **Turbo XT** monokróm monitorral, 102 gombos billentyűzettel. Cím: Belme Attila, 2030 Érd, Fácán köz 3/4.

Eladó **IBM Turbo XT**: 20 MB HDD, 360 kB FDD, 4S/2P/1G, CGA + Hercules vezérlőkártya, 14"-os Hercules monitor, 102 gombos billentyűzet, GM6 egér, Centronics printer. Irányár: 60 000 Ft. Cím: Kelemen Menyhért, 9023 Győr, Ifjúság krt. 92.

Eladó új, garanciális **Amiga 500** 1 MB-ra bővítő, gyári órával, kapcsolóval, RF modulátorral reális áron. Cím: Oláh Lajos, 3014 Hort, Kossuth út 147.

VESZEK

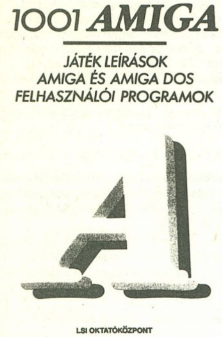
Keresek **C-64-re** 2.0-ás vagy bármilyen más verziójú Geos-t, valamint Geos segédprogramokat (lehetőleg angol nyelvűeket). Kapcsolatba lépnék IBM gépen dolgozó, programozási nyelvek iránt érdeklődőkkel! Cím: Szabó Péter, 5100 Jászberény, Korányi út 1.

CSERÉLEK

Enterprise programokat cserélek 5,25"-os és 3,5"-os lemezeken is. Cím: Veress Zoltán, 4244 Újfehértó, Hársfa u. 29.

1001 Amiga

Játékleírások. Amiga és Amiga DOS. Felhasználói programok. (Budapest, 1990. LSI Oktatóközpont, 154 oldal. Ára: 253,— Ft)



A magyar Amiga-tulajdonosok sajnos nincsenek túl jól ellátva szakirodalommal, ami a programozókra és felhasználókra egyaránt vonatkozik. Éppen ezért vehettük örömmel a kezünkbe az LSI Oktatóközpont 1001 Amiga című könyvét, mely a közkedvelt „1001-es" sorozat legújabb tagja. Az Amiga-tulajdonosok széles körének érdeklődésére tarthat számot: a játékosok a mikrolexikonnak és a játékleírásoknak, a programozók pedig a Seka és a Devpack közkedvelt assemblerfordító és a Diskmaster lemezkezelő program ismertetésének örülhetnek. Mindkét csoport számára fontos lehet az AmigaDOS és a Turboprint II. leírása. Főleg a kezdőknek hasznos az Amiga hardverfelépítésének ismertetése, valamint az Amiga vírustenyészetének és irtásuknak a leírása. Mindent összevetve, a kiadvány az „1001"-es sorozat jól bevált szerkezetét tükrözi, egyaránt segít a kezdőknek és a gyakorlott programozóknak. A könyv egyetlen árnyoldala a terjedelem. Az érintett témakörök közül néhányról lehetett volna kissé bővebben is írni, ezért mondhatjuk, hogy kissé referenciakártya jellegű az egész. Azok, akik a sorozat régebbi, nyolcbites gépekről írt tagjait szívesen forgatták, de idővel áttértek a korszerűbb Amiga számítógépre, ezt a kötetet is hasznos segítségnek tekinthetik.

G. A.

Ferenczy Imre — Gerő Judit:

Quattro

(Budapest, 1991. Számalk, 104 oldal. Ára: 235,— Ft)

Túlméretezett referenciakártyára emlékeztet ez a könyv. Szerzői valószínűleg nem kezdőknek szánták, hiszen a Quattro használatához szükséges alapvető információk hiányoznak belőle. Például nincsenek benne a program indításával kapcsolatos tudnivalók, és az sem, hogy a menürendszert a "/" billentyű lenyomásával lehet aktivizálni.

Az első részben — a menüfa struktúrájának megfelelően tagolva és hexadecimális osztálykódokkal ellátva — tömören ismertetik a menüparancsokat. Ezután a Quattro funkció-, kurzormozgató és egyéb különleges billentyűinek használatát írják le. Ezt követi a Quattro függvényeinek bemutatása, típus szerint csoportosítva.

A könyvnek közel felét a makrók használatával kapcsolatos tudnivalók foglalják el. Részletes ismertetést találunk az egyes makróparancsokról, példákkal fűszerezve. A makrókban használható speciális billentyűnevek és a menüekvivalens parancsok felsorolása zárja a makrókról szóló részt. Zavaró, hogy a billentyűfunkciókra itt csak röviden utalnak, folyton vissza kell lapozni a billentyűtáblázathoz. A menüekvivalens parancsok táblázatának használatánál is sokat segítene a menüparancsok leírásánál alkalmazott kódszám.

Az egész kötet az összecsapottság benyomását kelti. A hiányzó információkról már szóltam, emellett jó néhány zavaró hiba is előfordul. Csak a legfeltűnőbbet említem: a 32. oldalon az utolsó „Quit-vége" hatása nem „vissza a táblázathoz", hanem kilépés a Quattro-ból, azaz vissza a DOS-hoz.

Összegezve: ennek a könyvnek egyetlen erénye, hogy bármikor kézbe vehető, fellapozható. De a számítógép mellett a Quattro sokkal több segítséget nyújt az F1 billentyű lenyomása után.

(bl)



BIBLIOGRÁFIA

E havi összeállításunkban olyan — a közelmúltban megjelent — számítástechnikai tárgyú könyvek között válogattunk, melyekről rovatunkban eddig nem jelent meg ismertetés.

Klár András — Babócsy László: Windows 3 mindenkinek. Budapest, 1991. Trafcomp Kft., 105 oldal. Ára: 298,— Ft.

Nyéki Lajos — Nagy Tamás: Turbo Basic. Budapest, 1991. LSI Oktatóközpont, 385 oldal. Ára: 453,— Ft.

Sályi János — Szelezsán János: Adatbázisok. Budapest, 1991. Számalk, 103 oldal. Ára: 282,— Ft.

Vonalban lenni

Amiről rovatunk e havi első darabja szól, a konkrét termék szintjén várhatóan csak az elkövetkező években jelenik meg a hazai — és a külhoni — piacon, a dolog jelentősége azonban feltétlenül indokolja, hogy már most szóljunk róla. A világ vezető számítástechnikai cégeinek e nagy horderejű, az elkövetkező évek fejlesztési irányát meghatározó bejelentése mellett azonban nem maradnak el a gyakorlati életből származó információk sem.

Az ACE nem áll meg!

A Montana cég nyár elején tartott szemináriumán az SCO képviseletében jelen levő Zbig Zdanowicz úr számolt be a számítástechnikai ipar 21 vezető vállalatának arról a közös szándékáról, hogy szabványokon alapuló, fejlett számítástechnikai környezetet (ACE = Advanced Computing Environment) akarnak kialakítani, melynek révén számítástechnikai rendszerek új osztálya hozható létre.

A tervek szerint az ACE két korszerű operációs rendszeren (Open Desktop és OS/2) alapul majd, a hardverre vonatkozóan pedig az Intel- és RISC-vonalat szánják dominánsnak. Tehát a választott operációs rendszerek egyike az SCO Open Desktop nevű UNIX-alapú grafikus operációs rendszere, amely több ezer UNIX-alapú felhasználói program működését támogatja már ma is, és használható a laptopoktól a multiprocesszoros servergépekig. A másik operációs rendszer a Microsoft cég új operációs rendszere, az OS/2 3.0, amely támogatja az MS-DOS, Windows, OS/2 és POSIX alatt futó felhasználói programokat.



Az ACE egyik hardverbázisán a szabványos x86-alapú személyi számítógépek és rendszerek támogatása azt jelenti, hogy a 386-os, 486-os és jövőbeli x86-os számítógépek továbbra is kitűnő alapot nyújtanak majd az igényes alkalmazásokhoz. A másik hardvervonalnak szánják az új szabványspecifikációjú, vagyis a MIPS cég R4000 RISC-processzorára alapozott számítógépeket.

Az ACE a már ismertett elképzelések mellett a már meglevő hálózatokhoz is biztosít majd csatlakozási lehetőséget. Vagyis a felhasználó a különböző gyártók által előállított számítógépeken egyaránt használhatja majd a már meglevő és az újonnan fejlesztett programjait. Így a PC-s környezetben használt fejlesztések nem vesznek el, ha a felhasználó a RISC-világba tér át.

Ugyanakkor a fejlesztőket is új alkalmazások írására ösztönzi az a tény, hogy az ACE-specifikációknak megfelelő szoftverek igen széles számítógépbázison lesznek majd futtathatók. Mivel a fejlesztők néhány ACE-termékhez már ebben az évben hozzáférhetnek, az első felhasználói programok megjelenését 1992-re várhatjuk.

Hogy a csík sikk legyen!

A Magics Kft. céljául tűzte ki, hogy a világ más országaihoz hasonlóan Magyarországon is az ipar és a kereskedelem minél több területén alkalmazzák a vonalkódtechnikát. Ebben nagy segítséget nyújt a ICS (Identcode System) nemzetközi szervezete, amelynek a Magics is tagja. Az általuk kínált vonalkódos eszközöket az ICS-közösségben rendelik meg nagy mennyiségben, ezért a legnagyobb viszonteladói kedvezményben részesülnek. Természetesen a kedvezmény hatása pozitívan érződik a hazai árakon is. Az ICS-tagság miatt lehetőség van arra is, hogy a most kialakítandó nemzeti szabványokat a nemzetközi szabványokhoz igazítsák. Ebben segít az ICS azzal is, hogy a magyar szakemberek állandó továbbképzését biztosítja. A vonalkódtechnikai rendszerek alkalmazásával meggyorsíthatjuk az adatok bevitelét, és



egyúttal lehetővé válik az információk biztonságos kezelése. Ugyanis a vonalkódtechnikával feldolgozott adatoknál a tévedés jóformán kizárt, mivel kb. 3-5 millió karakterenként fordulhat elő tévedés, mind bevitelnél, mind pedig megjelenítésnél.

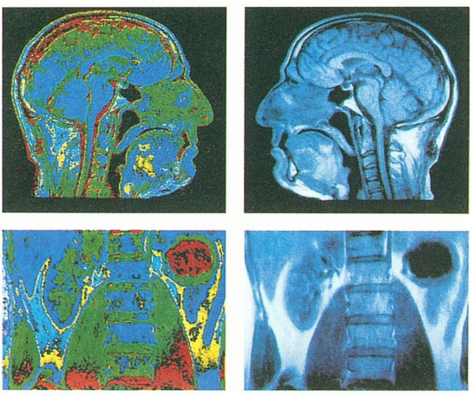
Vásárlásaink során már sokan láttuk, hogyan olvassa le egy érintőscanner vagy egy lézerolvasó a vonalkódtechnikai címkével ellátott árukat. Tulajdonképpen a pénztárgépek írható/olvasható memóriakártyájára írják rá a különböző árukra vonatkozó információkat. Így erre a kártyára könyvelik rá az egész napi anyagmozgásokat. Zárás után a memóriakártya kiemelésével napi készleteket, kimutatásokat, táblázatokat készíthetünk. Ennek a folyamatnak a technikai megvalósítását végzi a Magics Kft. Nagy választékban értékesítenek különböző vonalkódolvasó perifériákat (lézerpisztoly, vonalkódolvasó ceruza), ipari nyomtatókat. A kereskedelem területén például jó minőségben és kedvező áron (kb. 100 000 forintért) forgalmaznak vonalkódnyomtatókat is. Természetesen a nyomtatókhoz tartozó címkék széles választékát (vegyszernek, az időjárás viszontagságainak ellenálló etiketteket) is megtaláljuk kínálatukban. A vonalkódtechnikai eszközök értékesítése mellett komplex alkalmazásokat is készítenek. A rendszerek létrehozásakor figyelembe veszik, hogy az alkalmazók nem a legdrágább, hanem az igényeiknek megfelelő, megbízható megoldásokat szeretnék. Rövid működésük alatt már jó néhány termékkövetési, minőségbiztosítási, raktári, kereskedelmi, szállítási, könyvtári rendszert hoztak létre, amelyek közül a legjelentősebb a Lehel Hűtőgépgyár termékkövető rendszere.

Békében — együtt

Júniusban ismét a Béke Szállóban tartotta szokásos nyári termékbemutatóját a Microsystem. Az elegáns környezetben a számítástechnika mellett — immár szinte természetesen — kiemelt szerephez jutott az irodatechnika. Itt a különböző típusú és teljesítményű másolók és faxok mellett a szép kiállítású telefonok, valamint a korszerű telefonközpontok vonták magukra az érdeklődők figyelmét. Nem hiányozhattak az irodák világához tartozó kiadványszerkesztő rendszerek, valamint az ügyviteli szoftverek sem. Különösen nagy volt a választék a különböző kis, közepes és diszkontáruházak készletnyilvántartó és anyaggazdálkodó rendszereiből.

A felhasználói szoftverek mellett hardvertermékeket is felvonultattak a bemutatón. Természetesen a Microsystem — mint a Compaq dealere — ismét bemutatta valamennyi Compaq gépét, a Novell rendszerintegrátoraként pedig ezt az alkalmat használta fel arra, hogy bemutassa a Novell 3.11-et 386-os gépekre. Megtudtuk azt is, hogy a teljes 32 bites multitasking hálózati operációs rendszer 282 000 forintért vásárolható meg.

Megint csak a Microsystem hagyományaihoz híven kiemelt szerepet kaptak az egészségügyi alkalmazások. A ma már széles körben használatos (kórházi, gyermekorvosi, körzeti orvosi) rendszerek mellett újdonságként szerepelt a „a házi orvosi" szoftver vonalkódos receptnyomtató modulja, valamint egy komplett terhesgondozási és szülőszobai programrendszer. Szintén itt mutatták be először azt az ultrahangvizsgálat-nyilvántartó szoftvert, amelyet egy képfeldolgozó rendszer támogat. Itt a képi adatfeldolgozást és adattárolást a Protomix Kft. által kifejlesztett grafikus kártya biztosítja. Segítségével az 1024x1024-es felbontású, 256-féle színárnyalatú képeket optikai lemezen tárolhatjuk tömörítve. A bemutatott konfiguráció (386/33 MHz alapgép, 150 Mbájtos winchester, VGA-monitor, grafikus képfeldolgozó kártya,



grafikus monitor, scanner, 1 Gbájtos optikai lemez, videofelvevő) meglehetősen drága, kb. 1,7 millió forint az ára.

A Microsystem vendégeket is hívott bemutatójára. Compaq gépeken a SWISSCAD Kft. mutatta be azt az általuk kifejlesztett AutoCad-alapú komplex építészeti tervezőrendszert, amely nagy sikert aratott a kiállításlátogatók körében. Ugyancsak vendégként volt jelen a soproni Magics Kft. is, amely már az Ifabón is kitűnt vonalkódtechnikai megoldásaival.

Név és cím a szerkesztőségben!

Az Alaplap 1991. júniusi számában adtunk hírt egy számítógépcsalád hazai bemutatkozásáról. Az azóta eltelt rövid idő alatt azonban a számítástechnikai piacon egy szingapúri cég által gyártott, másik gépcsalád is megjelent. A két márkanév a megtévesztésig hasonlít egymásra — írásmódjuk mindössze egy magánhangzóban tér el egymástól, a kiejtésben ez a különbség is eltűnik —, holott a két számítógépgyártónak semmi köze sincs egymáshoz. Nem zárható ki tehát, hogy éppen a megtévesztés volt a cél, a hazai piacon később jelentkezett termékek a bevezetett márkanév „oldalvizén" kívántak révbe érni. Ezt a gyanút megerősíteni látszik, hogy az utóbbi termékek lényegesen olcsóbb, de egészen más műszaki színvonalú termékcsaládot reprezentálnak.

A közelmúlt eseményei is azt látszanak igazolni, hogy az összetévesztés elősegítése nem állt meg a stratégia szintjén. Egy dél-magyarországi városban jelentős méretű megrendelésre vonatkozó árajánlatkérésnél versenytársként indult a két cég. A piacon később jelentkezett, olcsóbb gépeket képviselő ismert nagyvállalat árajánlatában csak kompatibilitási szintet jelölt meg, típusszámot nem közölt, és termékeit konfigurációnként közel 40 000 forinttal olcsóbban kínálta. Az ártárgyaláson a vevő a típusszámmal ellátott, magas műszaki színvonalú gépeket választotta ki, de úgy vélte, hogy ugyanazokat kapja meg olcsóbban a nagyvállalattól! Mert hogy végül a nevek összetévesztése folytán — jót derülve? — az utóbbi vállalat szerezte meg az értékesítés jogát.

A tényekhez tartozik még, hogy a már említett városban nemrég megrendezett kiállításon a tudatos hitelrontásra és dezinformálásra utaló további jeleket is tapasztalhattak a kiállításlátogatók.

Rövid híradásunkkal arra szeretnénk felhívni a felhasználók figyelmét, hogy nagyon figyeljenek oda: vásárláskor a két hasonló nevű termékcsalád mely tagját választják, nehogy egy szándékukkal nem megegyező minőségű terméket vegyenek meg akaratuk ellenére!

Úgy gondoljuk, hogy a különböző árkategóriájú és eltérő színvonalú termékek jól megférhetnének egymás mellett a hazai piacon, ha a nevek hasonlóságát tudatosan nem használnák fel összekeverésükre.

Sziebig Andrea

Ha a megbízhatóság a döntő...

A MITAC 17 éves információipari hátterével
a technológia egyik távol-keleti vezetője.
Igen szigorú minőségbiztosító rendszerének
és hatalmas kutató-fejlesztő beruházásainak
eredményeképpen termékei a világ 65 országában váltak
a korszerűség és a megbízhatóság szinonimájává.

A megbízható gyártó termékei csak megbízható forgalmazó
tevékenysége nyomán képesek
a felhasználó javát szolgálni.

Ezért esett a MITAC választása hazánkban
az INTERAG-ra.



**VIGYÁZAT! Jól bevezetett és hírnévnek örvendő márkanevünkkel
kétes minőségű, hasonló hangzású nevek élnek vissza!**

Forgalmazó:

INTERAG

INTERAG INFORMATIKA
Budapest 1136 Pannónia u. 11.
Tel./fax.: 132-9375 Molnár Péter, Sugár Mihály

People Committed To InfoTech

MiTAC

ALPHA MICROSYSTEMS
AMERIKAI CSÚCSTECHNOLÓGIA
MAGYARORSZÁGON



CREATIVE COMPUTER SOLUTIONS

INFORMÁCIÓKÉRÉS: 19 ▲